International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 05 | May 2023 www.irjet.net p-ISSN: 2395-0072

CauseVCare - A Blockchain based Charity DApp

Parth Mangalkar[1], Piyusha Patil[2], Manjusha Katkhede[3], Assistant Prof. Sonal S. Fatangare[4]

[4]Assistant Professor, Computer Engineering, RMD Sinhgad School of Engineering, Warje, India
[1,2,3]UG Student, Dept. of Computer Engineering, RMD Sinhgad School of Engineering, Warje, India

---

Abstract - *A blockchain-based charity tracking system is a decentralized platform that uses blockchain technology to increase transparency, accountability, and efficiency in the charitable sector. The platform creates a public ledger that records all transactions and activities associated with charitable organizations, including donations received, funds disbursed, and expenses incurred. Smart contracts are integrated into the platform to ensure that the terms and conditions of transactions are automatically enforced, which increases trust and accountability for donors. Blockchain technology ensures that all transactions are immutable, transparent, and secure, reducing the need for intermediaries and eliminating associated costs.*

The blockchain-based charity tracking system improves donor confidence in charitable organizations, increasing donations and having a more significant impact. It provides insights into how funds are utilized and helps prevent fraud and corruption.

The blockchain-based charity tracking system is an innovative solution that can transform the charitable sector by promoting transparency, efficiency, and trust between donors and charitable organizations.

***Key Words*: Blockchain, Distributed Ledger Technology, Cryptocurrency, Transparent, Smart Contracts.**

1.INTRODUCTION

Most likely, we have seen that the problem in charity management is that donations are not reaching the needy; they are getting corrupted in between without knowing anyone because of a lack of transparency between donors and charities. Another thing is that the security of the funds and their related data have to be provided. A system that gives transparency and security to donations is needed. Here, we have a solution for this issue is, that we can use Blockchain for storing all data related to the donations.

Blockchain is a decentralized and distributed digital ledger that records transactions and stores data in a secure, transparent, and tamper-proof way. It was originally introduced in 2008 as the underlying technology behind Bitcoin but has since been applied to a wide range of applications beyond digital currency like healthcare, insurance, e-voting systems, identity verification, real estate, etc.

There are three types of blockchain available: public (Ethereum, Bitcoin), private (Hyperledger Fabric), and consortium (Ripple, IBM), and the types are classified on the basis of permissions to access the records and their authentication. We select them according to our needs for the application.

Blockchain gives transparency because anyone can access all transactions and verify them, as each block has a unique signature, timestamp, and reference to the previous block. Smart contracts are used to increase transparency, as they are self-executing programs used to trigger some actions when some conditions are satisfied. Proof of Work (PoW) or Proof of Stake (PoS) are the two consensus mechanisms used for the verification and validation of blocks before adding them to the blockchain. With this process, immutable records are made in the database. Smart contracts and PoS/PoW work together to provide a secure, decentralized, and automated system for validating transactions, executing contracts, and maintaining the integrity of the blockchain.

2. LITERATURE SURVEY

Sr no.	Name of Journal/Year of Publication	Paper Title	Author Name	Research Gap	Algorithms Used
1.	INT-JECSE - 2022	CrowdFunding Fraud Prevention using Blockchain.	Dheeraj Kumar S, Subash I, ShanthaKumari A, Deepa R	The authors have used the Proof of Work algorithm which is less time efficient as compared to its rival consensus algorithms such as Proof of Stake(PoS) and Delegated Proof of Stake(DPoS).	Proof of Work
2.	IRJET - 2022	Charity System using Blockchain Technology	Rhythm Negi Blessy Thomas, Prajkta Ghorpade Ammu Attiyilya	The authors use the PoW consensus algorithm and make use of Bitcoin blockchain which can only undergo 7 transactions per second.	Proof of Work & ECDSA
3.	IRJET - 2022	Blockchain Based Charity System Using PHP/MySQL	Varsha Kamble, Sapna Mandavkar, Hrishikesh Ramane	The authors have used Laravel for backend and PHP which is not suitable for modern Web applications which are based on the Blockchain technology.	Proof of Work
4.	Elsevier	Blockchain-based donations traceability framework	Abeer Almaghrabi, Areej Alhogail	The authors make use of the Bitcoin Blockchain which is comparatively slower as compared to the Ethereum Blockchain when it comes to the rate at which transactions are processed.	SHA-256 Bitcoin-P2P protocol
5.	IJRASET	Transparent Charity System using Smart Contracts on Ethereum using Blockchain	Purva Deepak Patil1 , Dikshita Jaiprakash Mhatre, Nidhi Hemant Gharat3 , Jisha Tinsu4	Future Work - MySQL will be used for centralized storage. In the paper authors haven't mentioned which algorithm they have used.	-
6.	International Journal of Research Publication and Reviews	Charity Donation System Based On Blockchain Technology	PROF.Dhanashri Patil, Abhishek Kadam , Gargi Sheytey , Tanmay Budage and Ashutosh Sonar	Not well chosen technology for building the application.	SHA-256.

7.	Elsevier B.V	DT-DPoS: A Delegated Proof of Stake Consensus Algorithm with Dynamic Trust	Yuanyuan Suna, Biwei Yanb, Yan Yaoc , Jiguo Yuc	The paper have told about the difference between DPoS and DT-DPoS , So there is no research gap.	Delegated Proof of Stake(Dpos) DT-DPoS
8.	Springer - 2021	A donation tracing blockchain model using improved DPoS consensus algorithm.	Xiujun Wang, Yufei Peng, Wei She	The authors have not given out a detailed layout of how to implement the proposed algorithm and make use of it in Decentralized Apps (DApps).	Delegated Proof of Stake and K-means algorithm
9.	Elsevier - 2020	A Blockchain-based CrowdFunding Platform for Future Smart and Connected Nation.	Vikas Hassija, Vinay Chamola, Sherali Zeadally	The authors have proposed the use of the PoVV algorithm. But the issue with this algorithm is that the energy consumption and other associated costs increase exponentially as the number of competing nodes (developers) increases.	Proof of Virtual Voting for showing relevant NGO suggestions to the donor. ECDSA for public key cryptography.

3. ALGORITHMIC SURVEY

Sr no.	Publication :	Algorithm Used :	Space/Time Complexity :	Remark :
1.	INT-JECSE - 2022	Proof of Work	Keccak-256 is used in a hash function which returns a 256 bits string or 32 bytes array. Time taken to add a new block - 12s.	Keccak-256 is stronger than usually used SHA-256.
2.	IRJET 2022	Proof of Work & ECDSA	ECC is significantly faster than the other counterparts like RSA which are used for public key cryptography. Time taken by ECDSA in signature generation and verification is 93ms & 125ms	Efficient algorithms like ECDSA and Keccak-256 are used making the overall process of encryption and exchange of keys very fast.
3.	IRJET 2022	Proof of Work	Keccak-256 is used in a hash function which returns a 256 bits string or 32 bytes array. Time taken to add a new block - 12s.	Keccak-256 is stronger than usually used SHA-256.
4.	Elsevier https://doi.org /10.1016/j.jksu	SHA-256,	Time complexity for 41 steps : 2253.5, (O(N))	The time and space complexities depends on the number of steps the algorithm has used,

	ci.2022.09.021		Memory requirement is 216 × 10 words O(1)	used widely by technology leaders.
5.	International Journal of Research Publication and Reviews	SHA-256.	Time complexity for 41 steps : 2253.5, (O(N)) Memory requirement is 216 × 10 words O(1)	The time and space complexities depends on the number of steps the algorithm has used, Used widely by technology leaders.
6.	Elsevier B.V	Delegated Proof of Stake(Dpos) DT-DPoS	Block generating time < 1 second Block generating time < 1 second	Improves the throughput of the transaction and verification speed. The number of witness node present in consensus are less so, Algorithms are more scalable. They are using ring signatures for more security.
7.	Springer 2021	Delegated Proof of Stake K-means algorithm	Block generating time < 1 second Time Complexity : O(N^2) (n is the input data size)	Improves the throughput of the transaction and verification speed. K-Means is slow when it comes to bigger datasets

4. PROPOSED SYSTEM

Our application consists of three major things, the frontend, the backend and the blockchain.

It further consists of two main applications:

1. The Charity application itself.

2. The application which will be used by the government where it will approve NGOs to be a part of the Charity Platform.

We have used Solidity for writing the Smart Contract for our DApp and used MySQL database for storing the registration details of the NGOs.

We are using Ethereum blockchain for storing all the transactions and data related to the NGO campaigns for safeguarding and preserving privacy of the NGOs while also maintaining transparency of transactions.

There are three types of users on our application and each one has different access rights to the app, these users are:

1. **Normal User:**

   Normal users can view campaigns and make donations to those campaigns.

   They can also search for specific campaigns to which they want to donate.

2. **NGO:**

   NGOs first register to the platform, and once approved by the government, they are capable of creating their campaigns.

   NGOs can create campaigns, update campaign details, and also delete their campaigns.

   They can also view the list of all the users who have made donations to their campaigns in real-time.

3. **Government (admin):**

   Admin will check the registration details and the proofs submitted by the NGOs during the registration.

   If the NGO seems legitimate, then only the admin will approve that NGO, and an approval mail will be sent to the NGO, after which that NGO can login to the Charity platform and create its campaigns.

   Else, a rejection mail will be sent to the NGO if the admin does not find the NGO to be trustworthy and rejects the NGO.

The smart contract forms the backbone for most of the functionalities provided by our application, and it helps in automating the process of transfer of money from the donor's wallet directly to the wallet of the intended NGO.

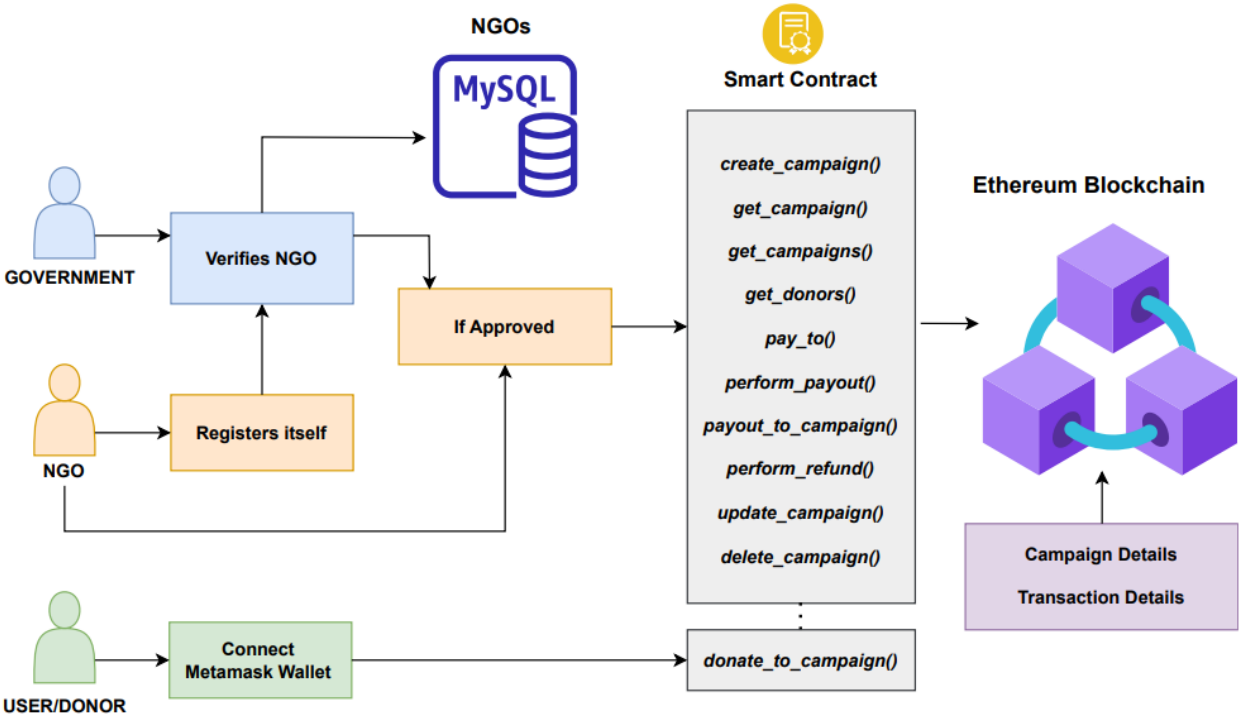


Fig. System Architecture.

Smart Contract:

A smart contract is a self-executing contract with the terms of the agreement directly written into code. It is deployed on a blockchain network, such as Ethereum, and is executed and enforced by the network's nodes. Smart contracts eliminate the need for intermediaries, as they automatically enforce the agreed-upon terms and conditions of an agreement. We created a smart contract for making transactions and making blocks of respective transactions to form blockchain and once the block is created it will not be modified as it is the blockchain block.

Contract Structure: The contract is named Charity.sol and has several state variables to store campaign and donation data. It facilitates charitable donations on the Ethereum blockchain. The contract has a campaignCount variable to keep track of the total number of campaigns created. The balance variable stores the total balance in the contract.

Mappings: Mappings are used to associate campaigns and donors with their respective data. campaignsOf maps an address to an array of campaignStruct, allowing users to retrieve campaigns owned by a particular address. donorsOf maps a campaign ID to an array of donorStruct, enabling retrieval of donors for a specific campaign. The campaignExists mapping is a boolean flag indicating whether a campaign exists based on its ID.

Events: The contract emits an event named "Action" to provide information about various actions performed within the contract and emitted whenever an action is performed within the contract. It provides information such as the campaign ID, the action type, the executor's address, and the timestamp.

Three structs are defined:

1.campaignStruct represents a campaign and contains information such as the campaign ID, owner address, title, description, image URL, cost, raised amount, deadline, number of donors, timestamp, and status.

2.statsStruct stores overall statistics, including the total number of campaigns, donors, and donations.

3.donorStruct represents a donor and contains the donor's address, donation amount, timestamp, and a flag indicating whether the donation has been refunded.

The statusEnum enum defines different statuses that a campaign can have, including "OPEN," "APPROVED," "REVERTED," "DELETED," and "PAIDOUT."

The contract includes several functions:

1. getCampaign(): Allows users to retrieve campaign details by providing the campaign ID.
2. getCampaigns(): Returns an array of all campaigns.
3. getDonors(): Retrieves an array of donorStruct for a specific campaign ID.
4. payTo(): It is an internal function that transfers funds from the contract to a specified address. It is used internally by other functions.
5. performPayout(): It transfers the raised funds of a campaign to the campaign owner. It updates the campaign status, emits an Action event, and adjusts the contract's balance accordingly.
6. payoutToCampaign(): Allows the campaign owner or the contract owner to initiate the payout process for a campaign.
7. performRefund(): Refunds the donations made to a campaign by transferring the donated amount back to each donor. It updates the donorStruct, adjusts the statistics, and emits an Action event.
8. createCampaign(): Allows users to create a new campaign by providing details such as the title, description, image URL, cost, and deadline. It adds the campaign to the campaigns array, maps the campaign to the owner's address, updates the statistics, and emits an Action event.

9. updateCampaign(): Enables the campaign owner to update the details of their campaign, such as the title, description, image URL, and deadline.

10. deleteCampaign(): Allows the campaign owner to delete their campaign. It updates the campaign status, performs refunds, adjusts the statistics, and emits an Action event.

11. donateToCampaign(): Enables users to donate to a campaign by providing the campaign ID and sending Ether with the transaction. It verifies the campaign status and donation amount, updates the campaign's raised amount and donor count, adds the donor to the donorsOf mapping, adjusts the statistics, and triggers payout or refund processes based on the campaign status and deadline.

Transaction:

Smart contract only interacts with blockchain to update, add new blocks with validation in blockchain, but blocks also have specific structure with which we can access the specific data at particular instant.

Block Structure of Blockchain:

Here we have eight parameters which are defining our block as follows: title of campaign, its description, status of donation, uploaded image while adding campaign, count of donors and owners to respective campaign, and raised funds for particular campaign.

The other main typical parameters are with which we can track the transaction and make secure decentralized structure as follows:

1.Sender: The address of the sender of the transaction.

2.Receiver: The address of the recipient or contract the transaction is intended for.

3.Value: The value or amount being transferred in the transaction.

4.Gas Limit: The maximum amount of computational resources (gas) allowed for executing the transaction.

5.Gas Price: The price the sender is willing to pay for each unit of gas used in the transaction.

To form a chain and ensure the integrity and immutability of the data stored within the blockchain, each block is connected by storing a hash of the previous block.

Ethereum Transaction Signing:

Ethereum utilizes the Elliptic Curve Digital Signature Algorithm (ECDSA) for authentication and transaction verification.

Key Generation: In Ethereum, each account is associated with a pair of cryptographic keys: a private key and a derived public key. The private key is a randomly generated, secret value that should be kept securely by the account owner. The public key is derived from the private key using elliptic curve mathematics.

Digital Signature: When a user wants to send a transaction on Ethereum, they construct the transaction and sign it using their private key. The signing process involves applying the ECDSA algorithm to the hash of the transaction. The result is a digital signature unique to that specific transaction.

Verification of Signatures: Other participants in the Ethereum network can verify the authenticity and integrity of a transaction by verifying its signature. To verify a signature, the recipient of the transaction or any network node can use the sender's public key and the provided signature. The verification process involves re-hashing the transaction and comparing it with the original hash derived from the signature. If the two hashes match, it means the signature is valid, and the transaction is considered authentic and unaltered.

Non-repudiation: The use of ECDSA signatures ensures non-repudiation, meaning that the signer cannot later deny their involvement in the transaction. Since the signature is unique to the transaction and derived from the signer's private key, it provides strong evidence of authenticity.

Protection Against Fraud: The use of ECDSA signatures protects against fraudulent transactions and prevents unauthorized individuals from sending transactions on behalf of others. Without the private key, it is computationally infeasible to forge a valid signature for a given transaction.

Overall, ECDSA in Ethereum plays a critical role in ensuring the security and integrity of transactions. It allows for authentication, verification, non-repudiation, and protection against fraud, contributing to the trust and reliability of the Ethereum network.

5. SYSTEM REQUIREMENT ANALYSIS

Following are the software requirements for building our Charity Dapp which are further divided into front-end, back-end and blockchain

IDE: Visual Studio Code.

Programming Languages: Solidity, JavaScript.

Libraries: ReactJS, Ether.js

Framework: Express, NomicLabs Hard Hat.

Database: MySQL

6. ALGORITHM DETAILS

1. **NGO Registration:**

I. The NGO enters the registration details through the registration form.

II. After submitting the registration details, the data will be sent to the backend and stored in the MySQL database.

III. The registration fields consist of:

a. NGO name. - string

b. NGO email. - string

c. NGO owner Identity or NGO registration proof. - file

d. NGO annual report. - file

e. NGO address – string

f. NGO password – string

IV. The NGO password will be encrypted using a Salt Hash of 10 rounds and a secret key which will be used for the password validation when the NGO tries to login using its registration details.

V. Server-side validation is done to check if email already exists in the database:

a. If email exists:

Appropriate message is displayed to the NGO

b. If email is valid and does not already exist:

Write NGO registration details to the database and send approval mail to the NGO through email.

2. NGO Login:

I. NGO enters and submits inputs corresponding to the fields in the login form.

II. Call is made to the back-end using API for verification of the inputs.

III. If NGO is registered:

a. And approved by the Government (admin);

Then send Authentication Token and user object to the front-end;

b. But not approved;

Then send an error header to the client application and show the appropriate error message to them.

IV. If NGO is not registered:

Return message "NGO must be first registered".

3. Create Campaign:

I. NGO enters the inputs in the fields required for creating the campaign, which consist of:

a. Campaign Title. - string

b. Campaign Deadline. - string

c. Amount to be raised. - integer

d. Image – file

e. Campaign Description – string.

II. After submitting details, users will be asked to pay some fee for publishing the campaign details to a block.

III. On submitting the details:

If the NGO pays the required gas fees, then:

a. The details will be appended to the campaign structure which was created in the smart contract which maintains the details of all the campaigns of all the NGOs.

b. Campaign details will also be appended to a Map data structure which helps to search for campaigns of specific NGOs with the help of the id of that NGO.

Else:

The transaction will be rejected, and the details will not be published to the blockchain.

IV. Once the campaign details are uploaded successfully to the blockchain, the NGO will be shown an appropriate success message on their screen

4. Update Campaign:

I. NGO enters the inputs in the fields required for updating the campaign, which consist of:

a. Campaign Title. – string

b. Campaign Deadline. – string

c. Campaign Description. - string

d. Campaign Image – file.

II. After submitting details, users will be asked to pay some fee for publishing the campaign details to a block.

III. On submitting the details:

If the NGO pays the required gas fees, then:

The details of the campaign will be updated directly in the campaign Struct with the help of the id of the campaign being updated.

Else:

The transaction will be rejected, and the details will not be published to the blockchain.

IV. Once the campaign details are updated successfully to the blockchain, the NGO will be shown an appropriate success message on their screen.

5. Delete Campaign:

I. NGO user clicks on the DELETE button.

II. On clicking the button, NGO will be reminded of their action and will be asked for a very small gas fee.

III. On paying the gas fee, the status of the campaign will be changed to "DELETED" in the campaign struct and the donations made to that campaign will be refunded to the respective donors.

6. Donate to Campaign:

I. The user will be asked to enter the amount that they want to donate.

II. On submitting the amount, the user will be asked to pay a minimal gas fee for the transaction to get reflected on the blockchain.

III. On entering the amount:

If the user pays the required gas fees, then:

a. The details will be appended to the campaign structure which was created in the smart contract.

b. The donors Structure will also be updated which consists of donations made to a specific campaign.

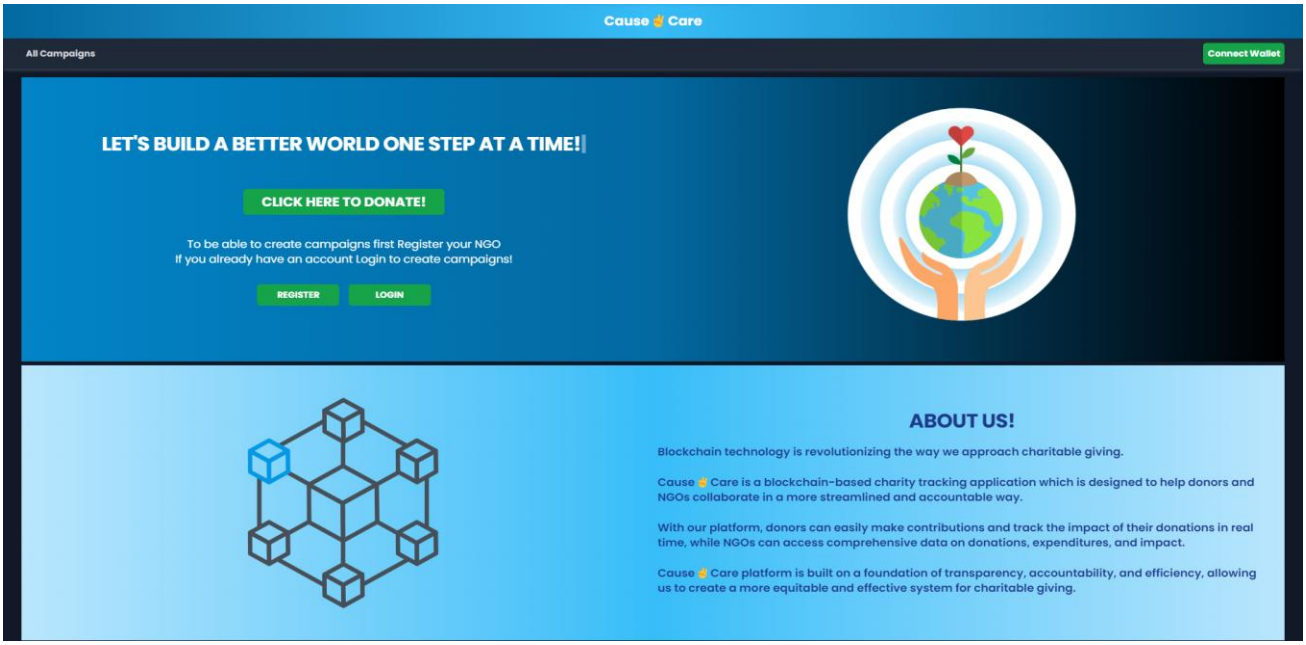
Else:

The transaction will be rejected, and the details will not be published to the blockchain, also the donation will not be made.

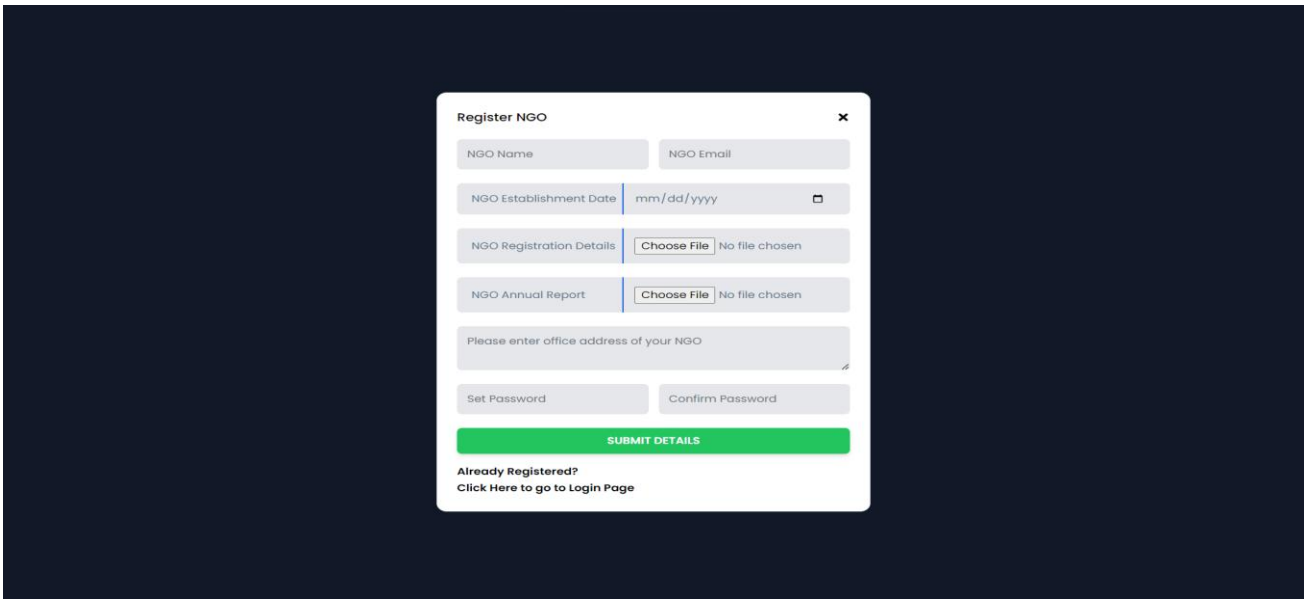
IV. On successful donation, the user will be shown an appropriate message on their screen, and the changes will be reflected throughout the blockchain.

7. RESULTS

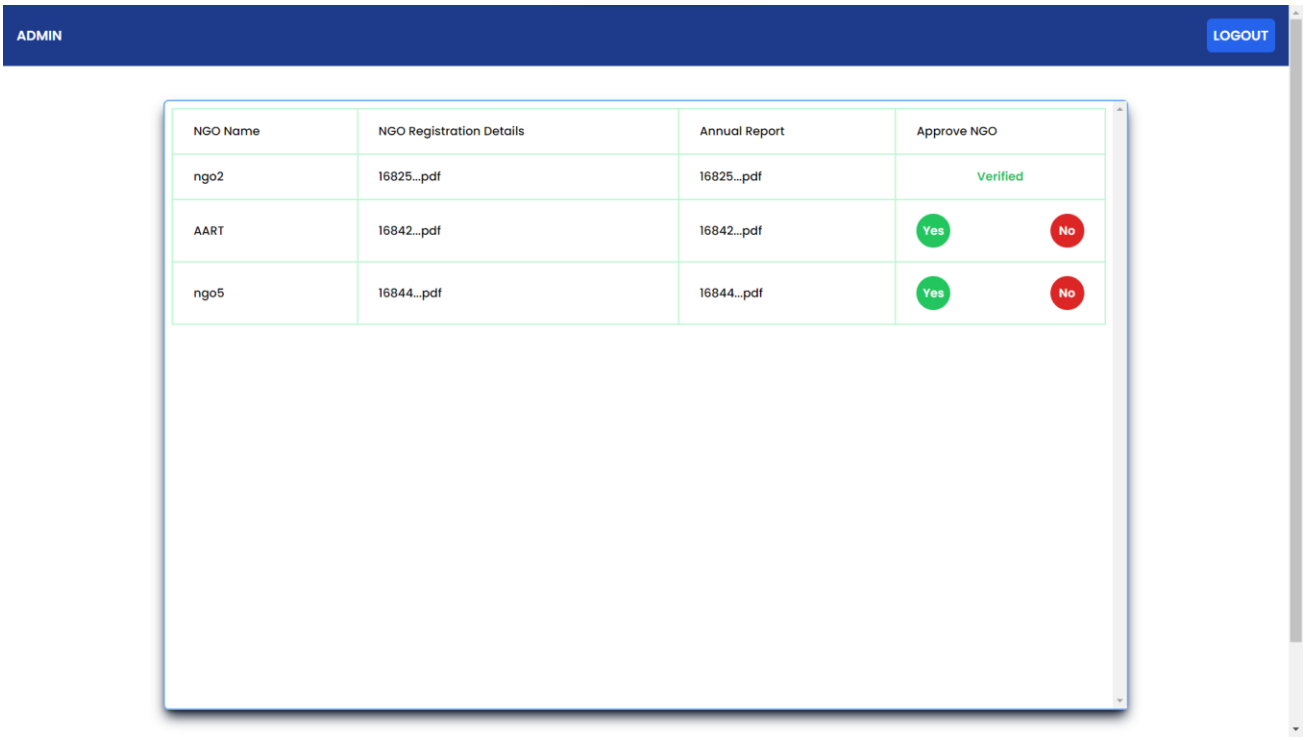
- Home page for normal users.
- Users can click on the "REGISTER" button if they want to Register their NGO.
- Alternatively, if an NGO is registered, then they can login as an NGO.
- Else, normal users can go to the Campaigns page for making donations to available campaigns created by the registered NGOs.



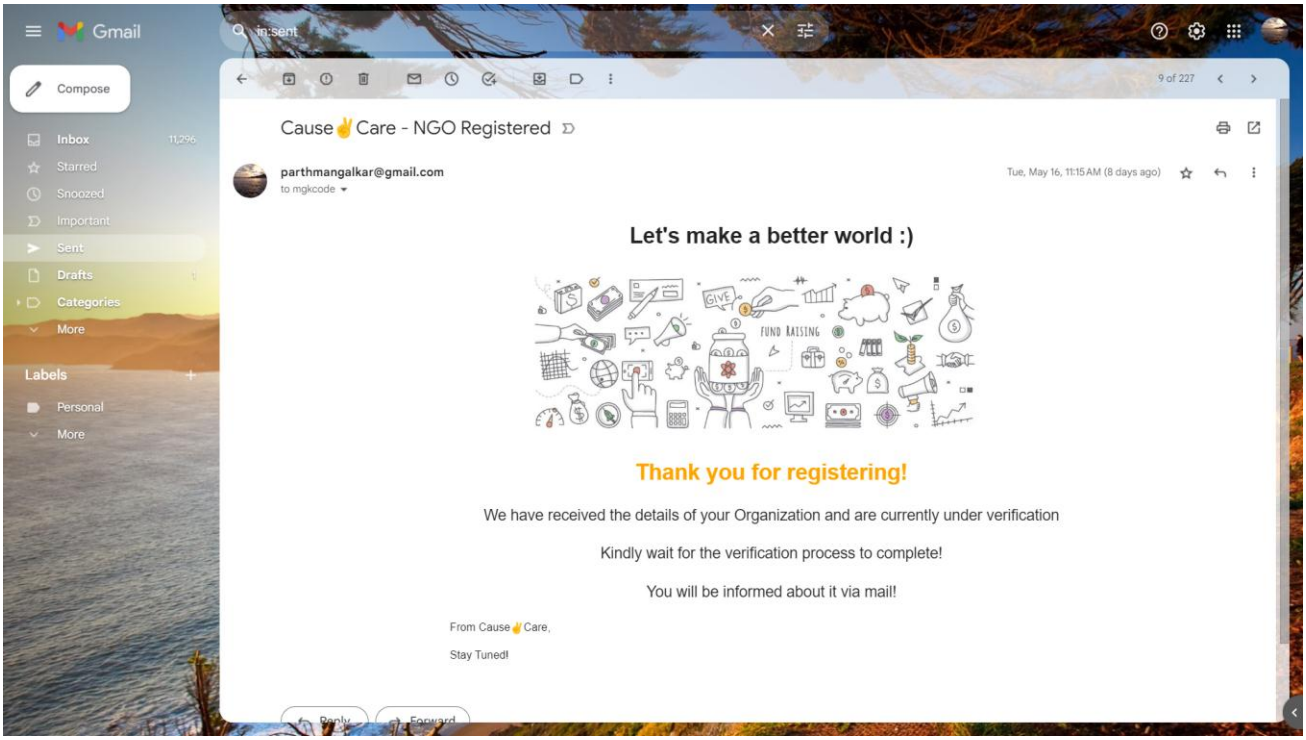
- Registration form using which NGOs can register themselves to be able to create campaigns.



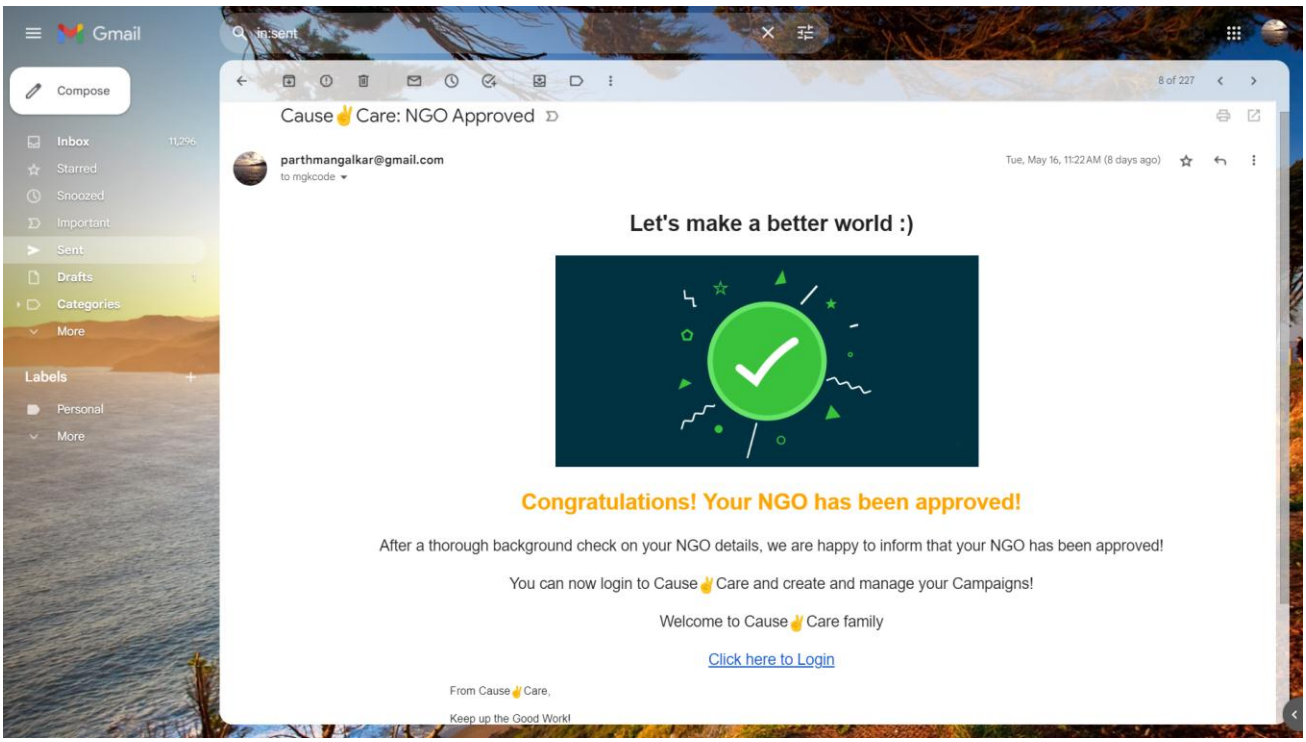
- Admin page where the admin (government) can view the details of the various registered NGOs.
- Admin can either approve or reject the NGOs.



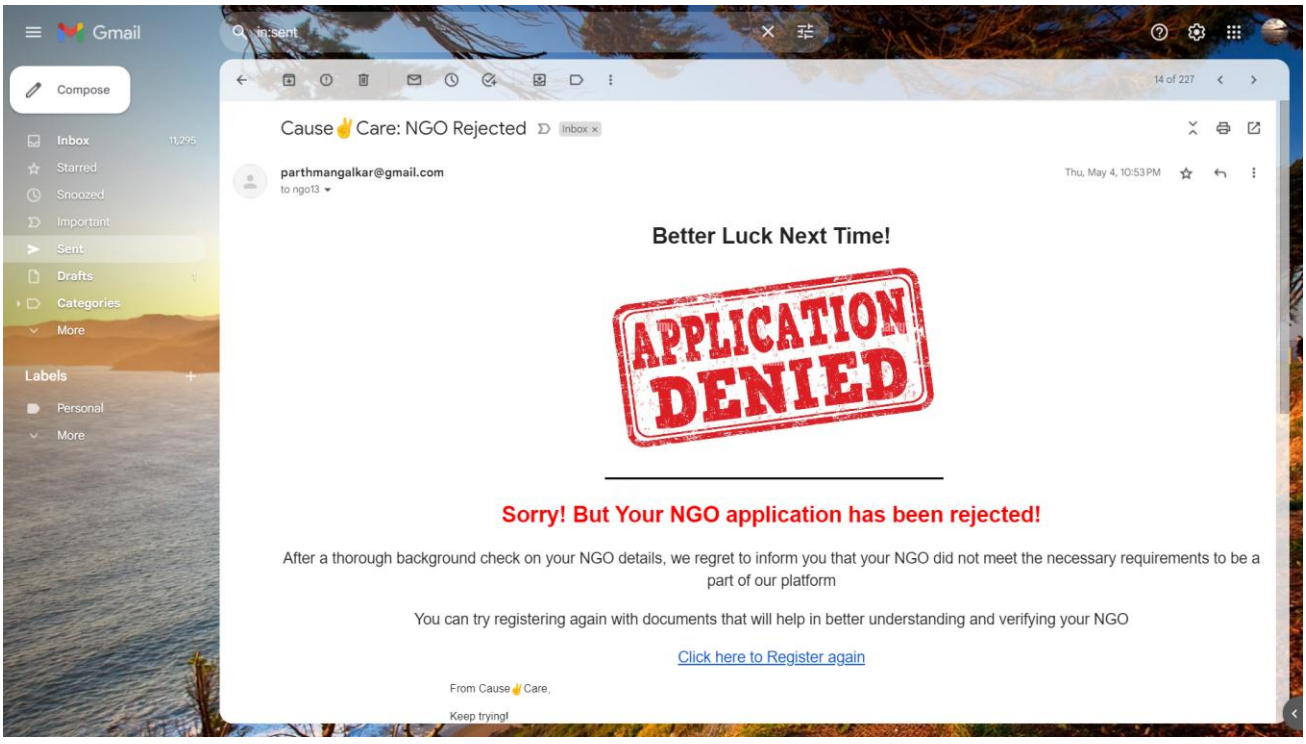
- Mail which will be sent to the NGOs on successful registration.



- Mail which will be sent to the NGO on approval by the admin (government)

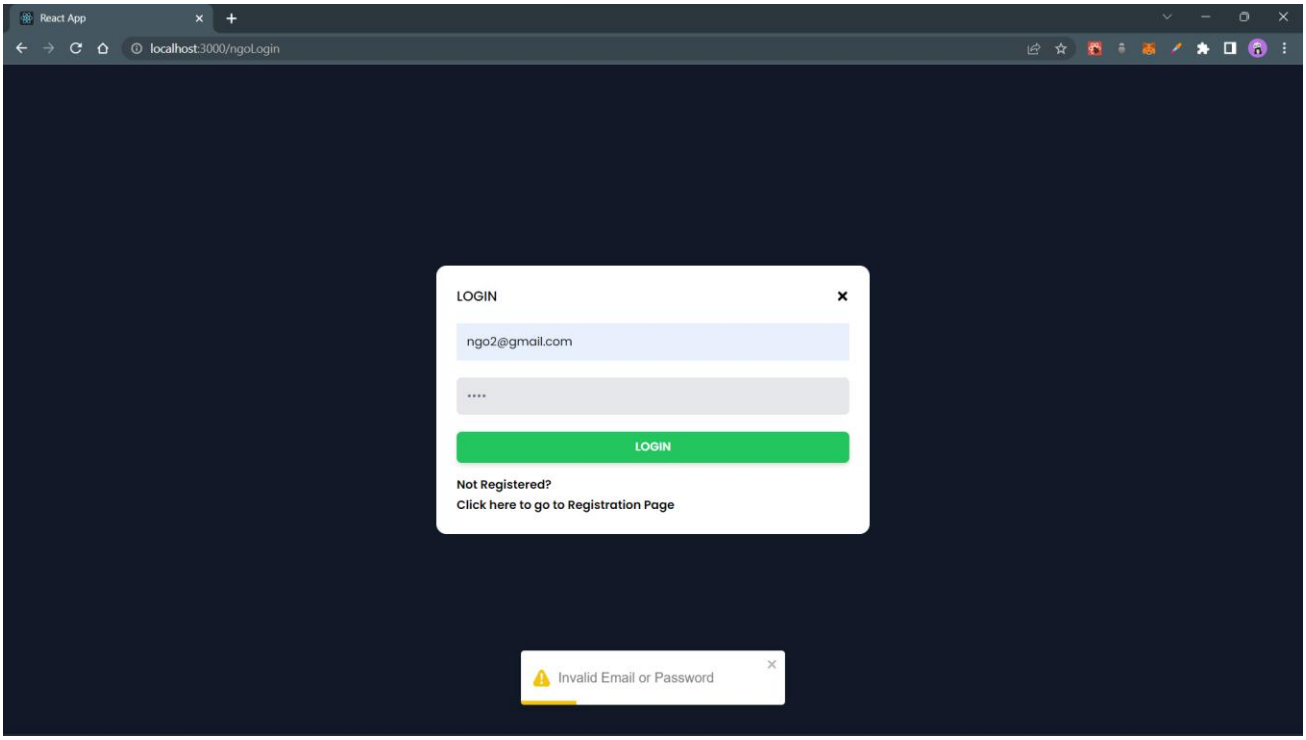


- Mail which will be sent to the NGO if their application was not approved by the admin (government)

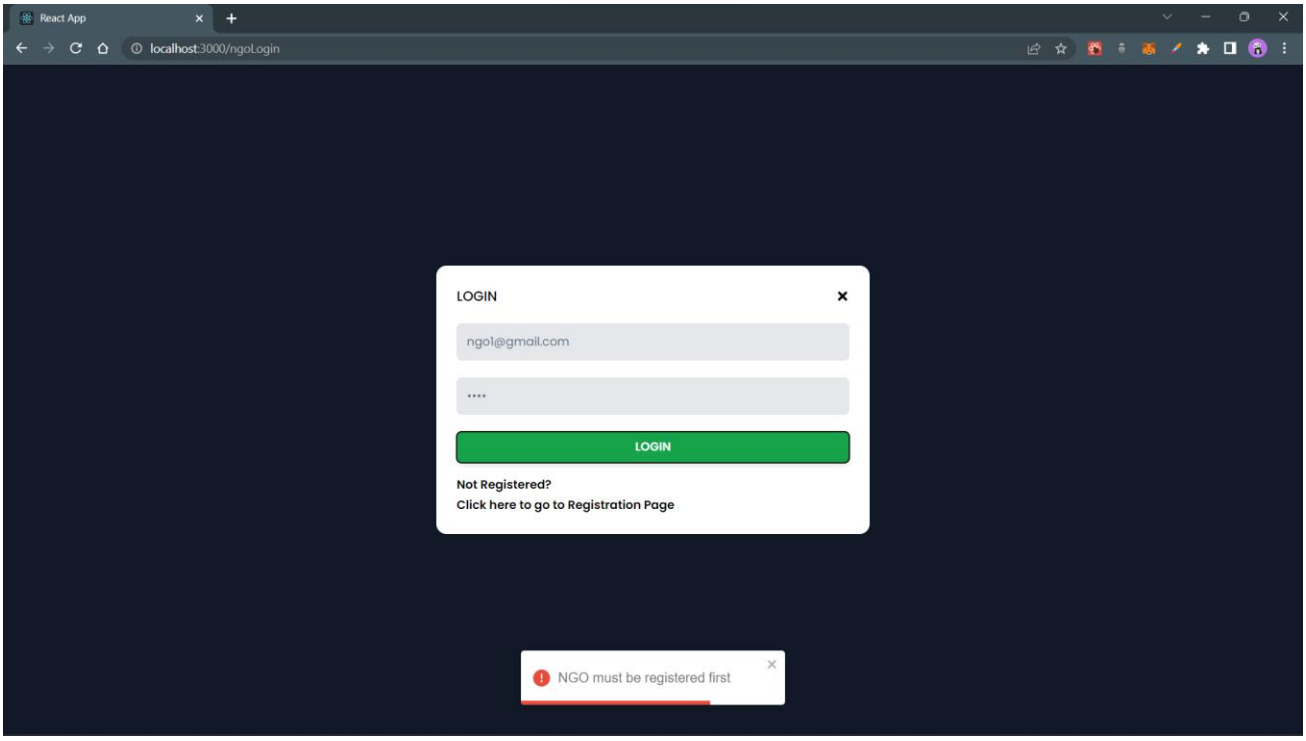


- NGOs can login only if their application was approved by the admin.

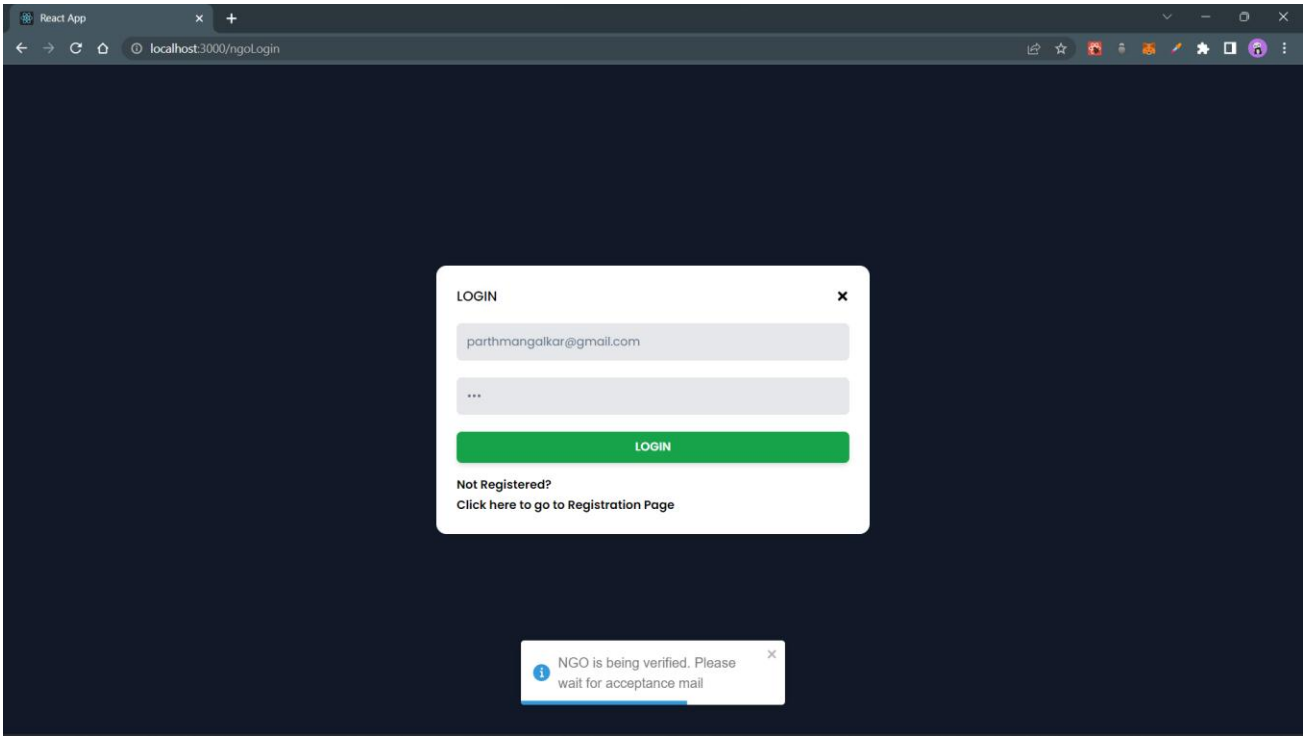
- If NGOs enter invalid credentials, then the following message will be shown using pop-up.



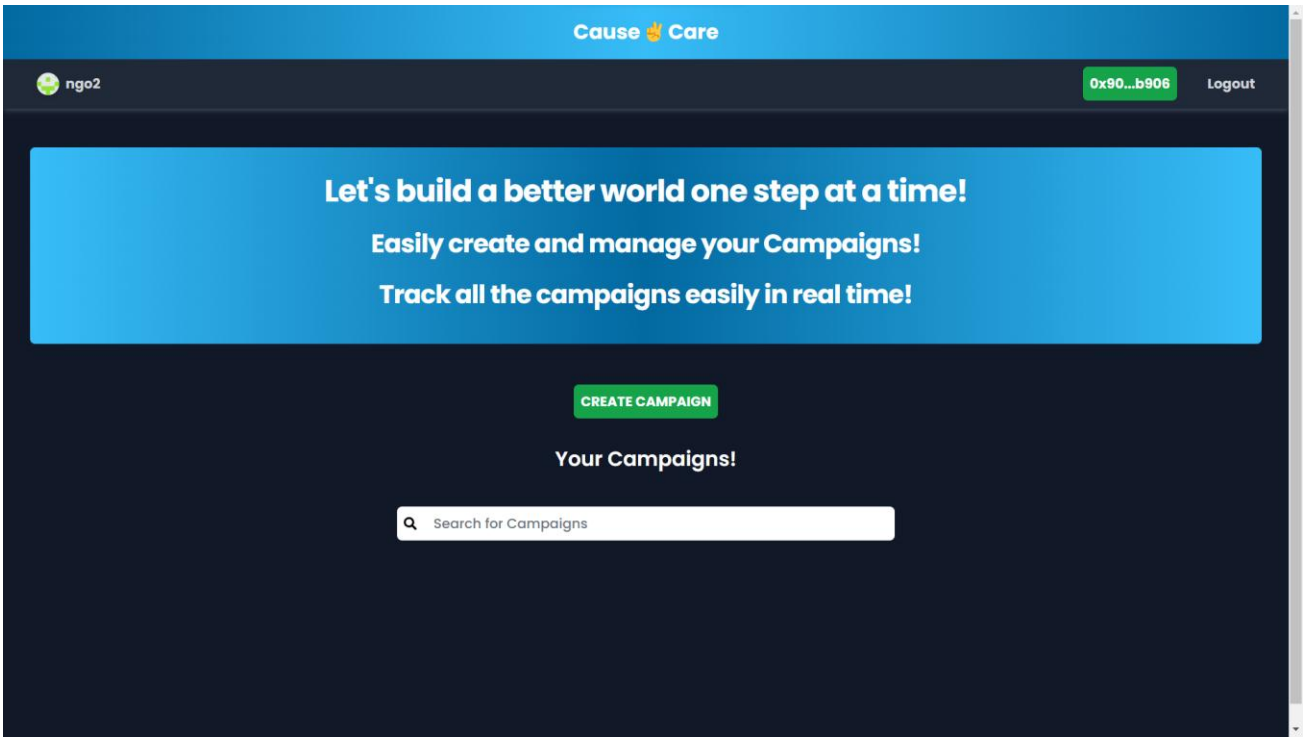
- If NGO tries to login when they haven't yet registered, then following message will be shown.



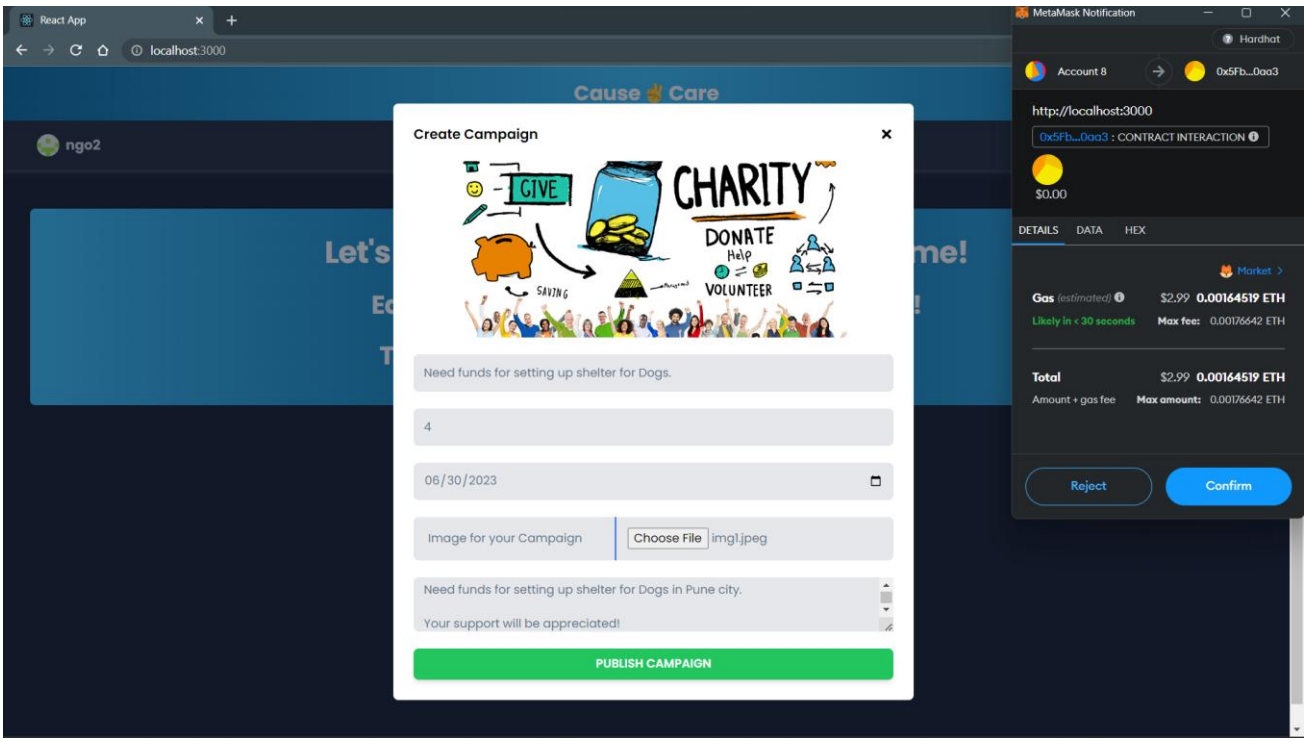
- If the NGO has registered but is still under scrutiny, then the following message will be shown.



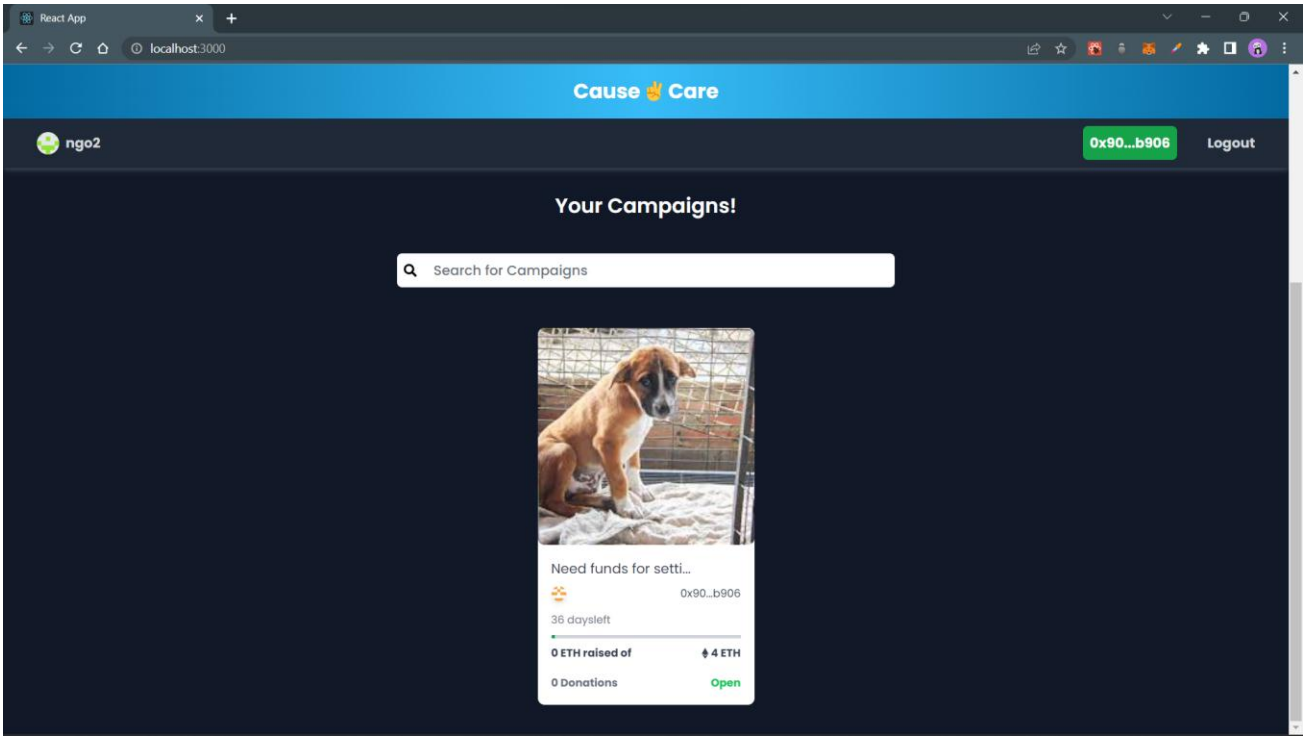
- Ngo home page



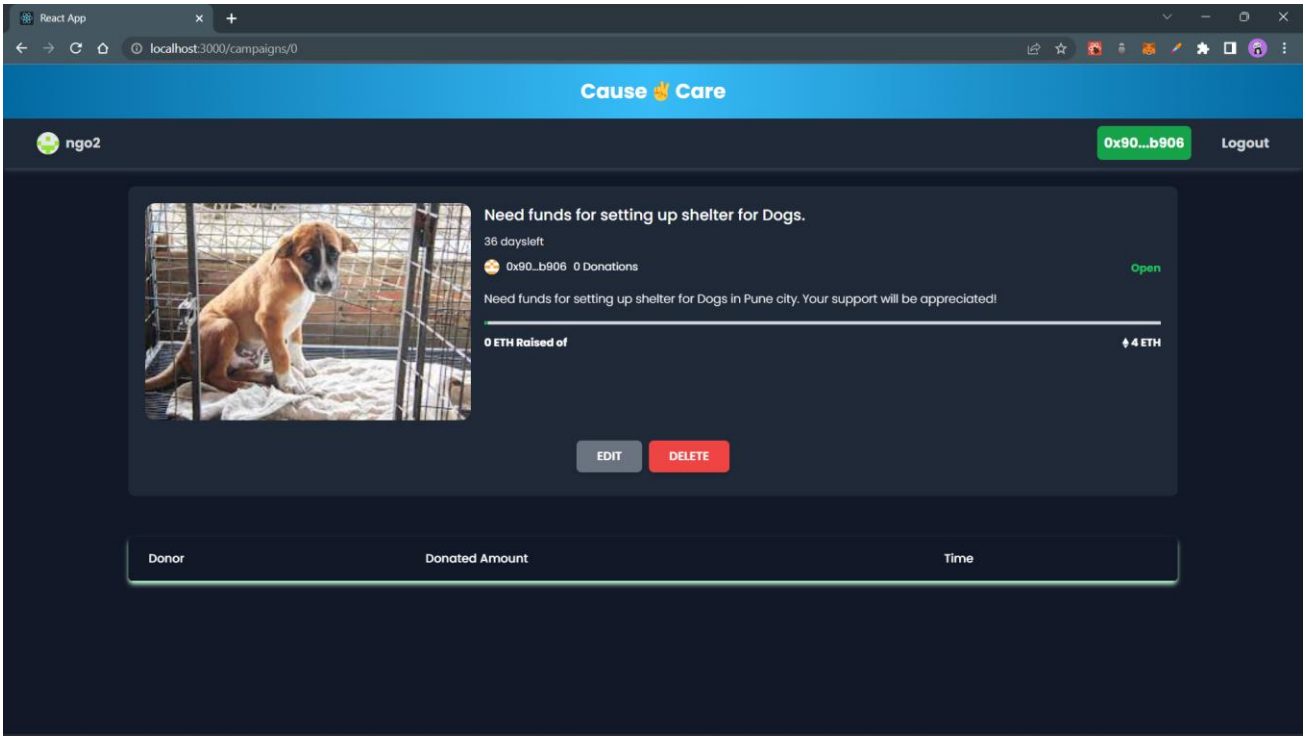
- Form to help NGOs in creating campaigns.



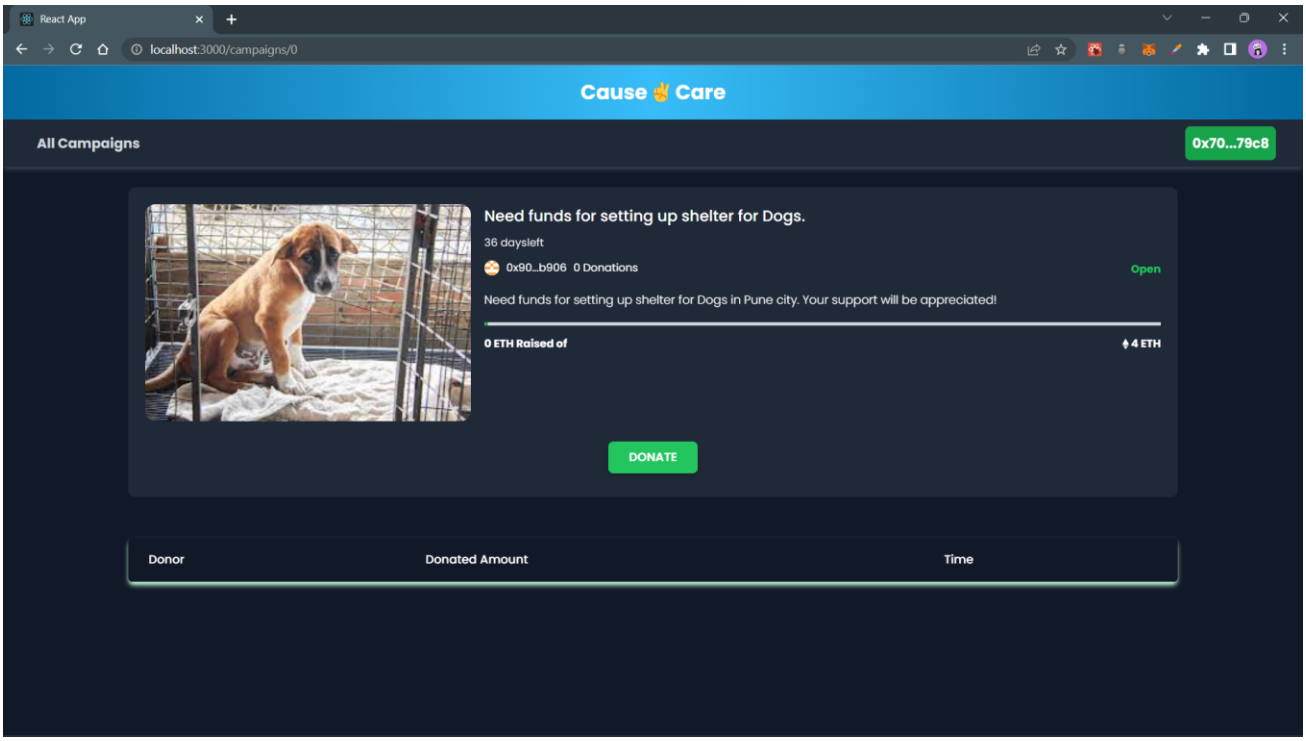
- Once the campaign is published successfully, it will be displayed on the Home Page.



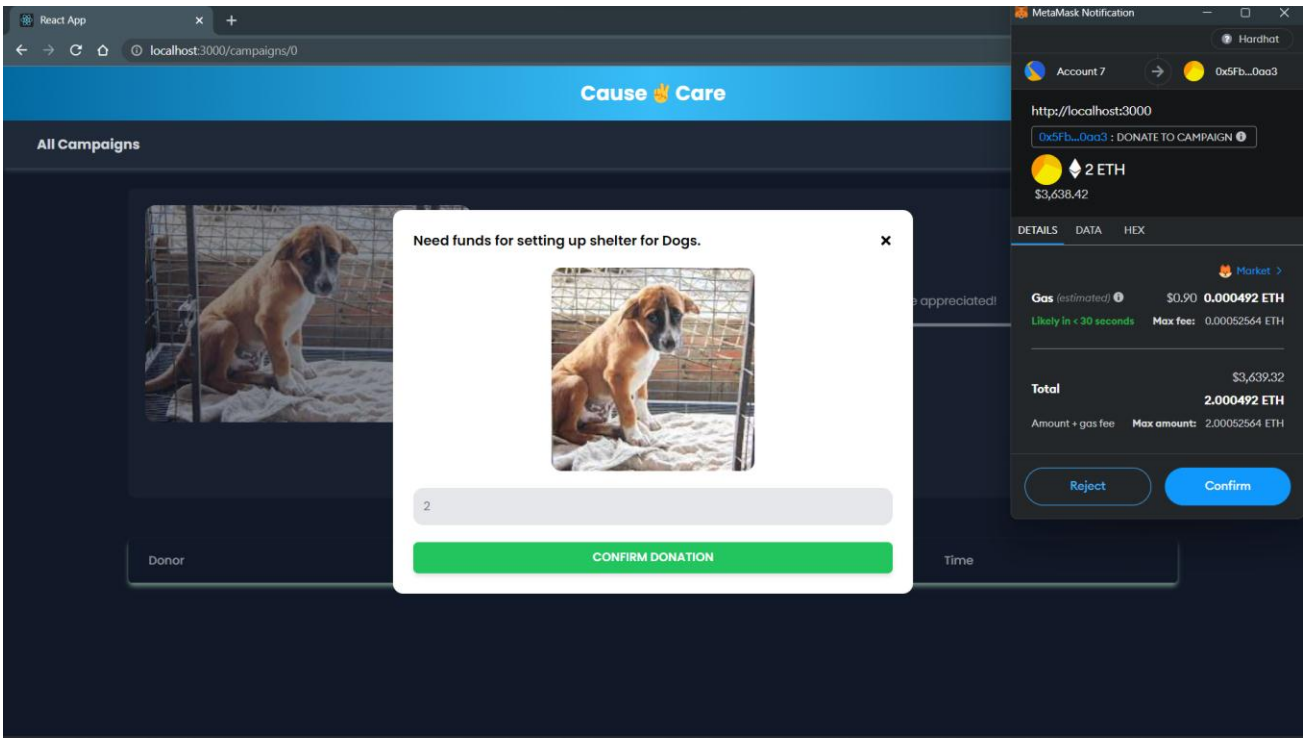
- Page where details of the Campaign will be displayed.



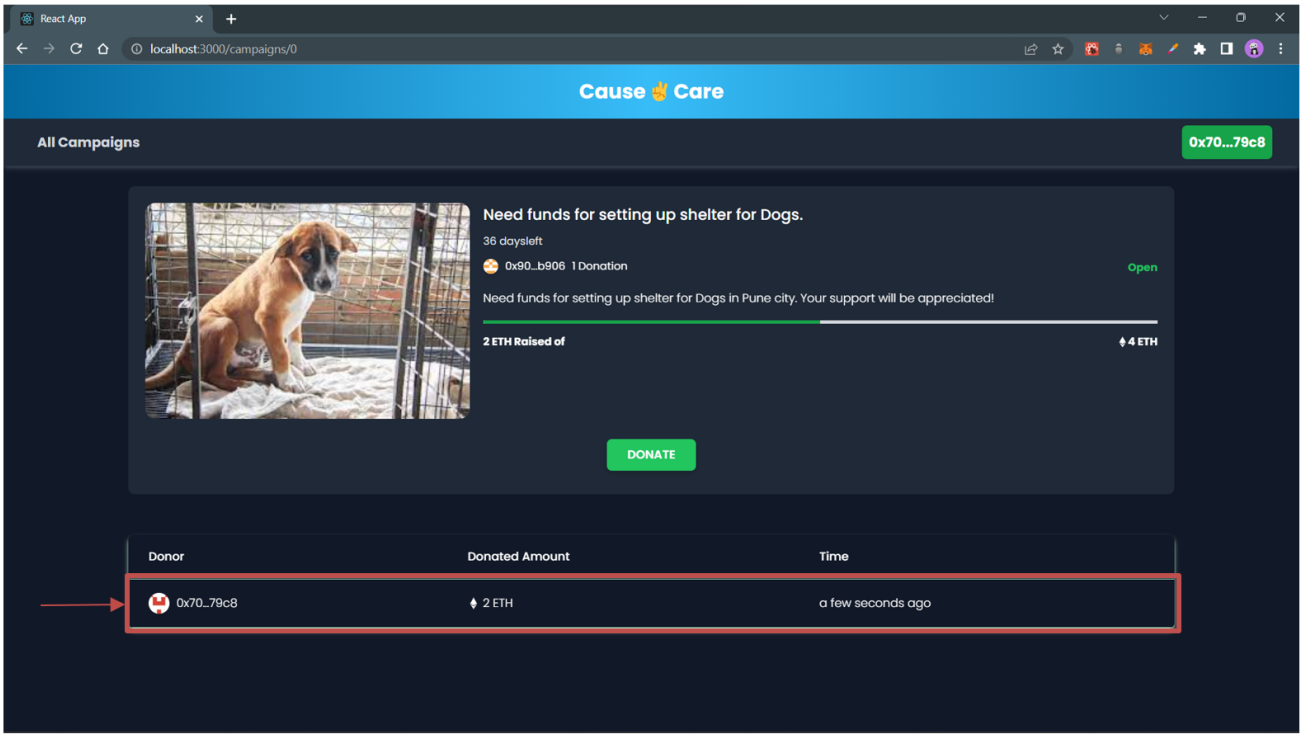
- Users can click on the “DONATE” button for making donation to the campaign.



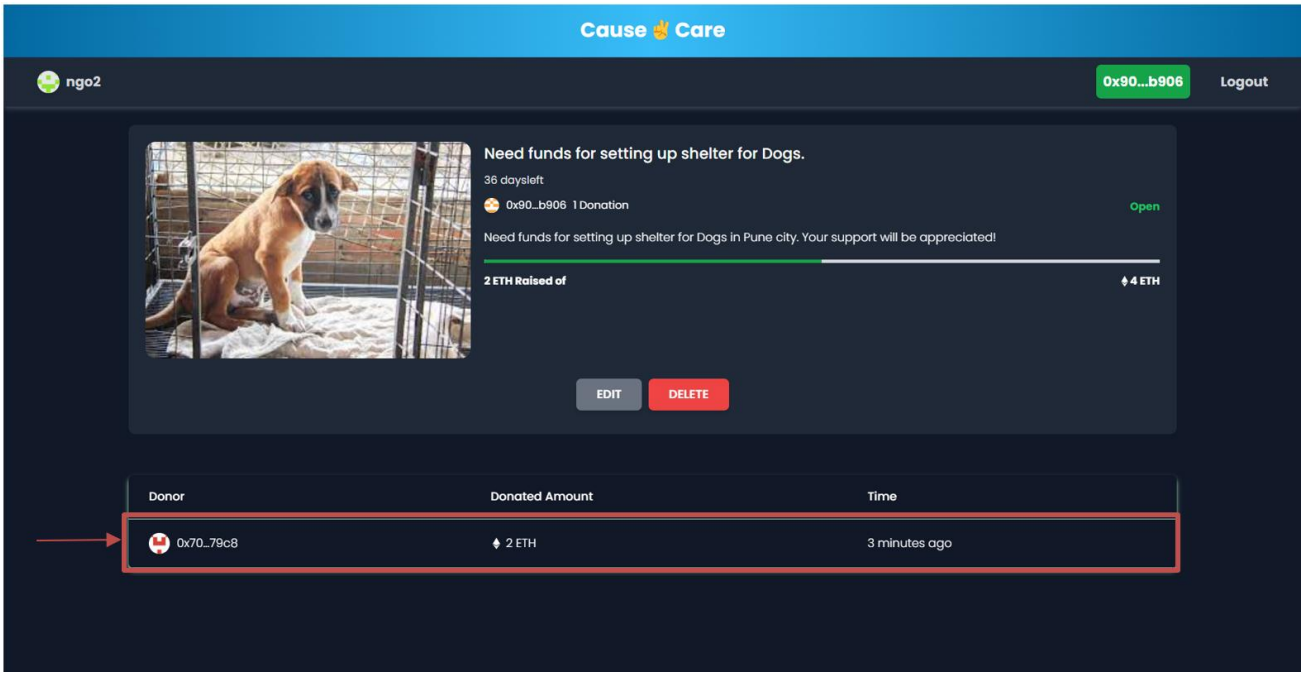
- Form which will be displayed on clicking the “DONATE” button.



- After the donation is made successfully, it will be displayed in the donors table as shown below.



- NGO can also see the donations that are made to their campaign.



8. CONCLUSION

In India, the current charity framework is plagued with issues such as low transparency, concerns around data security, lack of trust among individuals, and fake foundations. To tackle these problems, this paper proposes a novel approach that utilizes blockchain technology to revolutionize the charity framework. Our blockchain-based charity applications will also ensure that there is transparency in the transactions process and also that the process is not controlled by any one authority.

REFERENCES

[1] S. Pandey, S. Goel, S. Bansla and D. Pandey, "Crowdfunding Fraud Prevention using Blockchain," 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2019, pp. 1028-1034.

[2] Rhythm Negi1, Blessy Thomas2, Prajkta Ghorpade3, Ammu Attiyil4, Prof. Y. I. Jinesh Melvin5 “Charity System using Blockchain Technology”2022 International Research Journal of Engineering and Technology (IRJET)

[3] Jayasinghe, D., Cobourne, S., Markantonakis, K., Akram, R. N., & Mayes, K. (2017, September). Philanthropy on the blockchain. In IFIP International Conference on Information Security Theory and Practice (pp. 25-38). Springer, Cham.

[4] King, S., & Nadal, S. (2012). Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. self-published paper, August, 19(1).

[5] YILDIRIM, İ., & Şahin, E. E. (2018). Insurance Technologies (Insurtech): Blockchain and Its Possible Impact on the Turkish Insurance Sector. Journal of International Management Educational and Economics Perspectives, 6(3), 13-22.

[6] Hu, B., & Li, H. (2020). Research on Charity Systems Based on Blockchain. Research on Charity System Based on Blockchain, 768(072020), https://iopscience.iop.org/article/10.1088/1757- 899X/768/7/072020.

[7] Nixon, R. (2009). Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites. Shroff Publishers & Distributors Private Limited- Mumbai. 4.

[8] Rangone, A., & Busoli, L. (2021, March). Managing charity 4.0 with Blockchain: a case study at the time of Covid-19. Managing charity 4.0 with Blockchain: a case study at the time of Covid-19, 18(01), 31. https://doi.org/10.1007/s12208-021-00281-8.

[9] Ming Li, Jian Weng, Anjia Yang, Wei Lu,Yue Zhang, Lin Hou, Jia-Nan Liu, Yang Xiang, Robert H.Deng, " CrowdBC: A Blockchain-based Decentralized Framework for Crowdsourcing"

[10] Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017, June). An overview of blockchain technology: Architecture, consensus, and future trends. In 2017 IEEE international congress on big data (BigData congress) (pp. 557-564). IEEE.

[11] Aashutosh Singh, Rohan Rajak, Harsh Mistry, Prachi Raut " Aid, Charity andDonation Tracking System Using Blockchain " ISBN: 978-1-7281- 5518

[12] Bayu Adhi Tama, Bruno Joachim Kweka, Youngho Park, Kyung-Hyune Rhee, "A Critical Review ofBlockchain and Its Current Applications", International Conference on Electrical Engineering and Computer Science (ICECOS) 2017 DOI:978-1-4799-7675-1/17