**Crowdfunding DApp**

Submitted in fulfillment of the requirements of the lab

**BLOCKCHAIN LAB PROJECT**

By

|  |  |
| --- | --- |
| **Atharva Mahalungekar** | **21101B0045** |
| **Devank Shinde** | **21101B0047** |
| **Aditya Upasani** | **21101B0052** |

Supervisor

**Prof. Vinita Bhandiwad**



**Department of Information Technology**

**Vidyalankar Institute of Technology**

**Vidyalankar Educational Campus**

**Wadala(E), Mumbai - 400 037**

**University of Mumbai**

**(AY 2023-24)**

# CERTIFICATE

This is to certify that the Mini Project entitled “**Crowdfunding DApp”** is a bonafide work of **Atharva Mahalungekar, Devank Shinde, and Aditya Upasani** submitted to the University of Mumbai in fulfillment of the requirement for the lab of **BLOCKCHAIN LAB PROJECT.**

### Prof. Vinita Bhandiwad

Supervisor

### Dr. Vidya Chitre Dr. Sangeeta Joshi

Head of Department Principal

# Mini Project Approval

This Mini Project entitled “**Crowdfunding DApp”** by **Atharva Mahalungekar (21101B0045), Devank Shinde (21101B0047)** and **Aditya Upasani (21101B0052),** is approved for the lab of **BLOCKCHAIN LAB PROJECT.**

## Examiners

**1………………………………………**

(Internal Examiner Name & Sign)

### 2…………………………………………

(External Examiner name & Sign)

Date: Place:

# Contents

### Abstract iv

### Acknowledgment v

### Introduction 1

* 1. Introduction
  2. Motivation
  3. Problem Statement & Objectives

### Literature Survey 3

* 1. Survey of Existing/Similar System
  2. Limitation Existing/Similar system or research gap
  3. Mini Project Contribution

### Proposed System 6

* 1. Introduction
  2. Architecture/ Framework
  3. Algorithm and Process Design
  4. Details of Hardware & Software
  5. Experiment and Results
  6. Conclusion and Future work

### References 11

**Abstract**

### This project presents a Crowdfunding DApp (Decentralized Application) that leverages blockchain technology to create a secure, trustless, and transparent environment for fundraising. Traditional crowdfunding systems rely heavily on centralized platforms, which introduce trust issues, high service fees, and the potential for fraud or fund mismanagement. In contrast, this system removes intermediaries and ensures that all transactions are verifiable on the Ethereum blockchain.

### The DApp allows users to create campaigns by specifying parameters such as the campaign title, description, target funding amount, deadline, and an image. Contributors can browse through active campaigns and donate using MetaMask wallet integration. The DApp features real-time progress bars, donor information visibility, and campaign filtering once targets are reached.

### Built using a modern tech stack including React.js, Solidity, ethers.js, and Thirdweb SDK, the system ensures modularity, responsiveness, and security. Campaign data and donation records are stored immutably on-chain, and frontend state is handled using React hooks and context for seamless user interactions. Additionally, donor contributions are displayed alongside timestamps to maintain full traceability.

### The primary goal of this project is to decentralize the crowdfunding process, giving control back to the users, increasing transparency, and reducing dependency on centralized authorities. This DApp proves that blockchain technology can be effectively applied to solve real-world financial challenges while offering a user-friendly and efficient platform.

### Acknowledgement

### We would like to express our heartfelt gratitude to our project supervisor, Prof. Vinita Bhandiwad, for his continuous support, guidance, and motivation throughout this project. His expert advice and timely feedback helped shape this Crowdfunding DApp into a successful application. We are equally thankful to our Head of Department, Dr. Vidya Chitre, for her encouragement and for providing us with the necessary infrastructure and academic environment to conduct this project.

### We acknowledge the contributions of various open-source communities whose libraries, documentation, and tools were crucial in building this decentralized application. The developers of React.js, Solidity, ethers.js, Hardhat, and the Thirdweb SDK have laid the groundwork that enabled us to implement our ideas effectively.

### Special thanks to our peers and friends who participated in our testing sessions, provided user feedback, and helped us improve the user experience of our application. Finally, we are immensely grateful to our families for their unwavering support and encouragement throughout our academic journey.

### This project has been a valuable learning experience, deepening our understanding of decentralized applications, blockchain architecture, smart contracts, and the importance of security and trust in digital financial systems.

### Introduction

### Introduction

Crowdfunding is a method of collecting funds from a large number of individuals to finance a project, startup, or cause. The traditional approach to crowdfunding involves centralized platforms that serve as intermediaries between donors and fundraisers. While these platforms have democratized access to funding, they also introduce issues such as high transaction fees, limited transparency, and control over fund distribution.

With the evolution of blockchain technology, decentralized applications (DApps) have emerged as powerful alternatives that remove the need for central authorities. These applications use smart contracts deployed on blockchains to automate and secure operations. The immutable and transparent nature of blockchain makes it ideal for trust-based systems like crowdfunding.

This report outlines the development of a decentralized crowdfunding platform using Ethereum blockchain. It highlights the technical stack used, the system architecture, and the features implemented to ensure a secure, transparent, and user-friendly experience for both campaign creators and donors.

### Motivation

### The ever-increasing use of digital platforms for commerce, education, and social initiatives has created a corresponding need for efficient and trustworthy fundraising methods. Traditional crowdfunding platforms such as Kickstarter and GoFundMe have helped democratize funding by allowing individuals and startups to raise money from the general public. However, these platforms are not without their drawbacks. They are centralized, charge significant service fees, and often control the flow and release of funds, which may lead to delays and mistrust.

### The advent of blockchain technology has opened the door to new models of digital trust. Its core principles—immutability, decentralization, and transparency—make it ideal for use cases where financial integrity is critical. Crowdfunding, being a domain where money from multiple contributors needs to be securely collected, verified, and disbursed, naturally aligns with the strengths of blockchain.

### Our motivation stems from the desire to create a crowdfunding system that:

### Empowers campaign creators and donors to interact directly without relying on a third party.

### Ensures complete transparency in how much money is raised, from whom, and when.

### Uses smart contracts to automate the collection and distribution of funds in a secure, trustless manner.

### Demonstrates how Web3 and blockchain tools can be practically implemented by students for real-world solutions.

### 1.3 Problem Statement and Objectives

**Problem Statement:**

The major issue with current crowdfunding mechanisms lies in their centralization. Platforms act as mediators between campaign creators and donors, creating opportunities for opacity, misuse, and excessive control. Contributors have no guarantee that the funds are managed transparently or that campaigns will meet their stated objectives. Furthermore, platform fees eat into the funds raised, reducing the efficiency of the process.

**Objectives:**

* Design and develop a decentralized application that allows users to create and donate to campaigns using the Ethereum blockchain.
* Eliminate centralized control by replacing platform intermediaries with smart contracts that enforce fundraising rules automatically.
* Integrate wallet-based authentication via MetaMask to ensure secure and verifiable identity for both donors and campaign creators.
* Provide real-time campaign updates including progress bars, donor lists, timestamps, and visual indicators.
* Ensure immutability and transparency by storing all critical campaign data on-chain.

### 2. Literature Survey

### Crowdfunding and decentralized technologies have individually gained traction in recent years. Several studies and projects have explored how blockchain can revolutionize traditional funding models. A key insight from the literature is that most centralized crowdfunding platforms, though effective, suffer from issues of trust, high platform fees, and lack of fund traceability.

### Blockchain-based Crowdfunding Systems:

### Research by Voshmgir (2020) emphasized that decentralized crowdfunding models using Ethereum smart contracts can significantly improve transparency and trust. These systems automate fund collection and disbursement based on predefined rules without requiring a third-party administrator.

### Challenges in Existing Models:

### Studies like those by Zhang et al. (2022) point to the vulnerabilities of centralized systems to fraudulent campaigns and platform manipulation. Users have little insight into how funds are managed, and refund policies are often obscure.

### Smart Contracts for Automation:

### The concept of using smart contracts for automating crowdfunding logic has been proven successful in several pilot projects. Smart contracts can enforce milestones, return funds when targets are not met, and provide donors with evidence of fund utilization.

### User Trust and Transparency:

### Literature from decentralized finance (DeFi) use cases indicates that systems with public transaction logs (such as Ethereum) inherently provide a layer of auditability, which increases user trust. The ability to see donation histories and recipient addresses is a key benefit not offered by traditional platforms.

### This literature survey concludes that while many blockchain-based crowdfunding models are still evolving, they show great promise. The primary research gap is in delivering real-world usable platforms that balance usability with decentralization. Our project addresses this by building a complete full-stack crowdfunding DApp tailored for student and community-level initiatives.

### 2.2 Research gap

### Despite numerous advancements in the field of decentralized applications and blockchain integration, the existing body of work reveals several gaps specific to crowdfunding systems:

### Limited Real-World Implementations: While many concepts and prototypes exist, few projects have achieved full deployment for end-users in academic or small business environments.

### Usability Concerns: Many blockchain-based platforms lack intuitive interfaces, making them difficult for non-technical users to navigate.

### No Refund Logic or Escrow Mechanism: Several platforms lack built-in refund mechanisms or milestone-based disbursements for campaigns that do not reach their goal.

### Lack of Decentralized Media Storage: Many systems rely on centralized servers for image or metadata hosting, which defeats the purpose of decentralization.

### Security Gaps: Inadequate validation and input sanitization in smart contracts can lead to vulnerabilities or attacks on the contract state.

### 2.3 Mini Project Contribution

This project contributes a fully functioning, user-friendly decentralized crowdfunding application that addresses multiple research gaps:

* **End-to-End Functionality:** The platform includes campaign creation, donation, real-time updates, and post-target filtering.
* **Smart Contract Based Trust Logic:** Funding logic is enforced through Ethereum smart contracts ensuring immutability and auditability.
* **User-Centered Design:** A React-based frontend ensures a responsive and intuitive interface with clean UI for better accessibility.
* **Security Considerations:** The platform implements address-based donor tracking, restricted access for campaign control, and basic contract-level validation.
* **Scalable Architecture:** Easily adaptable for deployment on testnets or Ethereum mainnet with future enhancements for decentralized file storage and identity verification.

### 3. Proposed System

### 3.1 Introduction

### The proposed system is a decentralized crowdfunding platform designed to overcome the limitations of traditional models by utilizing Ethereum smart contracts. It allows users to launch fundraising campaigns and receive donations in a decentralized and secure environment. Smart contracts enforce campaign logic automatically, and all transactions are permanently recorded on the blockchain. This eliminates the need for a trusted central authority and promotes user control, financial transparency, and auditability.

### Users can create campaigns with metadata such as title, description, funding target, deadline, and image. Donors can browse campaigns, contribute funds using MetaMask, and view real-time progress. The system also features live donor history, campaign filtering, and automatic hiding of completed campaigns. It is designed to be modular, intuitive, and scalable for deployment on public Ethereum networks or within academic institutions.

### 3.2 Architecture/Framework

The system follows a full-stack DApp architecture with the following components:

* **Frontend (Client-Side):** Built using React.js, it handles all user interactions and dynamic rendering of campaign data. Tailored with reusable components, it manages user inputs, campaign lists, donation forms, and routing.
* **Blockchain Layer:** Smart contracts written in Solidity are deployed using the Hardhat framework to simulate a local Ethereum environment. MetaMask is integrated for wallet authentication and transaction signing.
* **Libraries and SDKs:**
  + ethers.js facilitates blockchain communication.
  + Thirdweb SDK simplifies interaction with smart contracts.
* **Data Flow:**
  1. User submits data from UI.
  2. React sends it to the contract via ethers.js.
  3. Smart contract executes logic (e.g., store donation).
  4. Events are emitted to update the frontend dynamically.
* **Security:**
  + Only the campaign creator can modify their campaigns.
  + Input validation is performed at both UI and smart contract levels.
  + Transactions are visible on the blockchain explorer for verification.

**3.3** **Modules**

### The platform is divided into five major modules:

### Campaign Creation Module:

### Allows users to create campaigns by submitting details like campaign name, description, image link, target amount, and deadline.

### Validates inputs and stores campaign data via a smart contract function.

### View Campaigns Module:

### Displays all active campaigns in a gallery-like interface.

### Shows campaign images, progress bars, deadlines, and donation targets.

### Filters out completed campaigns to focus user attention on active opportunities.

### Donation Module:

### Enables users to connect their MetaMask wallet and donate any amount in ETH.

### After donation, the system updates the total collected funds on the blockchain.

### Contributions are linked with timestamps and wallet addresses.

### My Campaigns Module:

### Displays campaigns created by the currently connected wallet address.

### Provides quick status updates and management overview for campaign owners.

### Campaign Detail View:

### Offers a deep dive into any specific campaign, including full description, donor list, total funds raised, remaining time, and target amount.

### Donor section includes timestamps and transaction hashes.

### 3.4 Details of Hardware and Software

### Hardware Requirements:

### CPU: Intel i5 or higher, 2GHz minimum

### RAM: 4GB (8GB recommended)

### Storage: Minimum 100GB

### Internet: Required for blockchain interaction

### Software Stack:

### Frontend: React.js

### Blockchain: Ethereum (Hardhat), Solidity, MetaMask

### Libraries: ethers.js, Thirdweb SDK

### Node.js & npm: Runtime and package management

### VS Code: Code editor

### IPFS (planned): For future decentralized file hosting

### 3.5 Experiment and Results

### To validate the proposed system, a series of functionality, usability, and performance tests were conducted.

### Functionality Testing:

### Successfully tested the creation and donation to multiple campaigns.

### Confirmed real-time progress updates after each donation.

### Verified automatic filtering of completed campaigns.

### User Testing:

### A set of peer testers reviewed the app interface.

### Feedback showed ease of use, clear layout, and fast navigation.

### Suggestions included adding search and sort filters for campaign lists.

### Error Handling:

### Ensured invalid inputs are caught (e.g., negative amounts, empty fields).

### Handled MetaMask connection failures with error prompts.

### Performance Testing:

### React pages load in under 1.5 seconds.

### Blockchain interactions (create/donate) complete within 10–15 seconds depending on network.

### Overall, the DApp performed reliably and met the design expectations.

### The results of these tests affirm the effectiveness of the Forecastly - Weather Dashboard in delivering a reliable and user-friendly weather forecasting solution.

### A screenshot of a computer Description automatically generated

### Fig 3.5.1 - Forecastly - Weather Dashboard for entire week

### A screenshot of a weather forecast Description automatically generated

### Fig 3.5.2 - Forecastly – Hour by hour weather of entire day

### 3.6 Conclusion and Future work

### Conclusion:

### This project successfully demonstrates how a decentralized application can improve upon the traditional crowdfunding model by ensuring transparency, trust, and direct user interaction. Through the use of smart contracts, campaign management and donations are automated and tamper-proof. The system is intuitive, secure, and effective in supporting peer-to-peer fundraising efforts.

### Future Work:

### Integrate decentralized image storage using IPFS.

### Enable campaign updates and milestone disbursements.

### Add user profiles with verifiable credentials (KYC).

### Explore DAO-based campaign verification for governance.

### Deploy to a live Ethereum testnet and conduct gas optimization.

### References

### Ethereum Foundation. (2023). Ethereum Documentation. https://ethereum.org

### Solidity Documentation. (2023). https://docs.soliditylang.org

### MetaMask. (2023). MetaMask Docs. https://docs.metamask.io

### Thirdweb SDK. (2023). https://portal.thirdweb.com

### Hardhat. (2023). https://hardhat.org

### ethers.js - The Ethereum JavaScript library. https://docs.ethers.org

### Voshmgir, S. (2020). *Token Economy: How the Web3 reinvents the Internet*. BlockchainHub.

### Zhang, Y., et al. (2022). A Review on Blockchain-based Crowdfunding Platforms. *International Journal of Financial Innovation*.

### Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* https://bitcoin.org/bitcoin.pdf

### Wood, G. (2014). *Ethereum: A secure decentralized generalized transaction ledger.* https://ethereum.github.io/yellowpaper/paper.pdf