CrowdChain 3.0: A Decentralized Milestone Trustless Escrow Crowdfunding Platform

Prof. Mr.Sanjaykumar Ranveer
Usha Mittal Institute of Technology
SNDT Women's University
Mumbai, India
su.ranveer@gmail.com

Amoli Dhuri
Usha Mittal Institute of Technology SNDT Women's University Mumbai, India
amolidhuri13@gmail.com

Nikita Jarad
Usha Mittal Institute of Technology
SNDT Women's University
Mumbai, India
nikitarjarad@gmail.com

Nikita Mane
Usha Mittal Institute of Technology
SNDT Women's University
Mumbai, India
Nikitamane793@gmail.com

Abstract—Centralized crowdfunding platforms face challenges such as high intermediary fees, limited transparency, and fraud risks. This paper proposes a decentralized crowdfunding framework using Ethereum-based smart contracts to automate campaign creation, milestone-driven fund disbursement, and refund mechanisms without third-party control. A comparative evaluation between Layer-1 Ethereum and Layer-2 solutions like Polygon demonstrates significantly reduced gas fees and improved transaction throughput, enabling cost-effective micro- investments. The proposed architecture enhances scalability, transparency, and security through structured smart contract design and vulnerability mitigation strategies.

Index Terms—Keywords— Blockchain, Decentralized Crowd- funding, Smart Contracts, Layer-2 Scaling, Web3 Security, Ethereum, Polygon.

INTRODUCTION

Crowdfunding has emerged as a revolutionary financial mechanism that enables entrepreneurs and individuals to raise capital from a global audience. However, contemporary crowdfunding platforms rely heavily on centralized Web 2.0 architectures such as Kickstarter and GoFundMe. These intermediaries impose high fees, operate with opaque policies, and act as single points of failure. Most critically, contributors lose visibility and control over their funds after contribution, leading to fraud and abandonment issues.

Blockchain technology offers a decentralized and trustless alternative. By transitioning crowdfunding to a distributed peer-to-peer ledger, transactions become cryptographically secured, immutable, and transparent. Smart contracts deployed on Ethereum automate escrow operations, ensuring that funds are only released if predefined funding goals and milestones are satisfied. If the campaign fails, contributors are automatically refunded without third-party intervention.

Despite these advantages, blockchain implementation introduces challenges such as high gas fees, scalability limitations, and smart contract vulnerabilities. This research proposes a multi-layered decentralized architecture integrating Layer-2

I. LITERATURE SURVEY

The integration of distributed ledger technology into financial ecosystems has prompted extensive academic research, particularly concerning the transition from centralized to decentralized crowdfunding models

A.Limitations of Traditional Crowdfunding

Early literature primarily focuses on the socioeconomic mechanisms of conventional platforms such as Kickstarter and GoFundMe. While these platforms successfully connect creators with global backers, researchers consistently highlight severe structural vulnerabilities, including opaque fund management, vulnerability to fraudulent campaigns, and exorbitant intermediary fees [**?**]. Furthermore, studies emphasize that once contributors transfer funds within traditional Web 2.0 systems, they lose jurisdictional control over the capital. This lack of post-funding accountability contributes significantly to declining investor trust.

scaling solutions to enhance performance while maintaining security guarantees.

B.Blockchain and Smart Contract Integration

To address the limitations of centralized intermediaries, recent scholarship explores blockchain-based architectures. Ethereum has emerged as the dominant framework due to its Turing-complete smart contract capabilities [**?**]. Researchers propose decentralized applications (dApps) in which smart contracts function as automated, mathematically enforced escrow mechanisms. These contracts securely lock funds upon contribution and release them only when predefined project goals are satisfied, thereby eliminating third-party reliance and minimizing the risk of asset misappropriation.

C. DAO-Based Governance Models

A prominent theme in contemporary research is the incorporation of Decentralized Autonomous Organization (DAO) governance mechanisms. Several proposed models introduce milestone-based funding protocols to ensure post-funding accountability [**?**]. In such systems, project creators cannot withdraw the entire treasury at once. Instead, they submit

structured spending requests for specific capital tranches. Contributors participate in a cryptographic voting process, and funds are released only upon majority consensus, effectively mitigating the risk of exit scams.

D. Scalability and Layer-2 Solutions

Despite the security advantages of decentralization, em- pirical studies frequently identify the scalability trilemma as a primary obstacle to mass adoption. Traditional Layer- 1 Ethereum deployments experience congestion during high network activity, resulting in slow transaction finality an elevated gas fees that render micro-investments economically impractical. Consequently, recent literature emphasizes the integration of Layer-2 scaling solutions such as Polygon. These protocols offload computational overhead from the Ethereum mainnet, significantly reducing gas costs and increasing transaction throughput while maintaining cryptographic security guarantees.

E.Identified Research Gaps

Although existing literature presents strong theoretical mod- els for decentralized crowdfunding, critical gaps remain. Many implementations prioritize functional deployment but lack comprehensive formal security verification against advanced smart contract vulnerabilities, including reentrancy and integer overflow attacks. Additionally, researchers highlight ongoing challenges in balancing transparent on-chain execution with off-chain storage optimization and user data privacy. Address- ing these limitations through optimized Layer-2 architectures and rigorous security auditing frameworks forms the primary motivation of the present research.

II. PROBLEM STATEMENT

While crowdfunding has successfully democratized access to capital, the contemporary ecosystem relies heavily on centralized Web 2.0 intermediaries that present significant structural flaws. The primary problem with existing crowdfunding models is that these intermediaries provide no deterministic guarantees or assurances to the investor once the capital is contributed. Investors possess no mechanisms to track how their funds are utilized or to enforce project deliverables, leading to a high prevalence of fraudulent campaigns, misappropriation of funds, and sudden project abandonment. Furthermore, traditional platforms levy exorbitant processing fees, act as single points of failure, and enforce opaque operational policies that ultimately deprive contributors of jurisdictional control over their investments.

Although blockchain technology offers a theoretical solution through decentralized ledgers and automated escrow, existing decentralized implementations face severe technical hurdles. Specifically, deploying smart contracts directly on Layer-1 networks such as Ethereum introduces prohibitive transaction costs (gas fees) and high latency during periods of net- work congestion, rendering micro-investments economically unviable. Additionally, immutable smart contracts are highly susceptible to catastrophic security vulnerabilities—such as reentrancy attacks and integer manipulation—which can lead to the irreversible loss of escrowed funds.

Therefore, the core problem this research addresses is the critical need for a secure, transparent, and economi- cally scalable decentralized crowdfunding architecture. The proposed system must eliminate reliance on centralized in- termediaries, guarantee post-funding accountability through automated milestone mechanisms, and resolve the economic friction of Layer-1 networks by leveraging Layer-2 scalability solutions, all while ensuring mathematically provable smart contract security.

III. BLOCKCHAIN-BASED CROWDFUNDING ARCHITECTURE

A. Core Concept

Blockchain-based crowdfunding operates on a decentralized peer-to-peer network that leverages distributed ledger technology and cryptocurrencies to facilitate trustless financial interactions. Unlike traditional centralized platforms, transactions are immutably recorded on a public ledger, ensuring transparency, auditability, and resistance to censorship. The elimination of intermediary control reduces administrative overhead while enabling deterministic execution of funding logic through smart contracts.

B. Multi-Layered Architecture

The proposed architecture follows a structured multi-layered blockchain model to ensure modularity, scalability, and security.

Data Layer: The data layer ensures transactional integrity using cryptographic hashing algorithms such as SHA-256. Each transaction is bundled into blocks, linked through hash pointers, and stored immutably across distributed nodes. This structure guarantees tamper resistance and verifiable historical traceability.

Consensus Layer: The consensus mechanism governs network agreement and transaction validation. The system utilizes a Proof-of-Stake (PoS) protocol to achieve energy-efficient block validation while maintaining strong security guarantees. Validators are selected based on stake weight, significantly reducing computational overhead compared to Proof-of-Work systems.

Application Layer: The application layer consists of Web3-enabled decentralized applications (dApps) that serve as the user interaction interface. Contributors and project creators access the platform through browser-based cryptocurrency wallets such as MetaMask, which manage private keys, sign transactions, and authenticate users in a non-custodial manner.

C. Smart Contracts as Escrow

Smart contracts function as autonomous escrow agents embedded within the blockchain. Upon contribution, funds are securely locked within the contract rather than being transferred directly to project creators. Release of funds is governed by predefined milestone approval logic and voting mechanisms. In cases where funding goals are not achieved or milestone requests are rejected, the contract automatically enables refunds to contributors. This deterministic and tamper-proof execution model ensures transparency, post-funding accountability, and investor protection without reliance on centralized oversight.

IV. SYSTEM ARCHITECTURE AND FUNCTIONAL MODULES

A. Application Structure

The platform utilizes a decentralized application (dApp) architecture, bridging standard web technologies with blockchain networks. The structure is divided into the follow-

ing components:

- **Frontend (Client Interface):** The user interface is developed using modern frameworks such as React.js or Next.js to ensure cross-device compatibility, scalability, and responsive design.
- **Web3 Integration:** The frontend communicates with the blockchain via Web3 injection libraries (such as Web3.js or Ethers.js). This enables the application to read blockchain data, interact with deployed smart contracts, and prompt users to authorize transactions securely.
- **Cryptocurrency Wallet:** Tools like MetaMask are integrated into the application. The wallet acts as a secure bridge, managing the user's private keys, authenticating identity, and cryptographically signing transactions without exposing sensitive credentials to centralized servers.
- **Backend and Testing Environment:** Node.js is used for optional server-side routing and API handling, while specialized development frameworks such as Hardhat or Truffle are utilized to compile, test, and deploy smart contracts onto the blockchain network.
- **Execution Layer (Smart Contracts):** The core business logic is written in Solidity and deployed to an Ethereum-compatible Virtual Machine (EVM). This decentralized execution layer securely stores campaign states, contributor mappings, and escrow balances without relying on traditional centralized databases.

B. Core Functionalities and Modules

The operational workflow of the platform is governed by algorithmic modules embedded within the smart contracts:

- **Campaign Creation Module:** A project creator initiates a campaign by invoking a creation function. The creator specifies immutable parameters including the campaign title, detailed description, target funding amount, minimum contribution threshold, and deadline. The smart contract records this data and generates a unique campaign identifier on the blockchain.
- **Contribution / Funding Module:** Investors browse active campaigns and contribute digital assets (e.g., Ether or MATIC) using connected cryptocurrency wallets. The funds are not transferred directly to the creator's personal wallet. Instead, the smart contract functions as an automated escrow, securely locking the funds and mapping the contributed amount to the investor's decentralized address.
- **Spending Request and Voting Module:** To prevent fund mismanagement, the platform incorporates a milestone-based approval mechanism. When a creator requires capital, they submit a spending request specifying the intended purpose and recipient address. Contributors to the campaign are notified and granted voting rights to approve or reject the request.
- **Finalization and Automated Resolution:** The smart contract automatically evaluates the voting outcome. If a predefined majority (e.g., greater than 50%) approves the request, the contract transfers the specified funds to the intended recipient. If the campaign fails to achieve
- its funding target before the deadline, or if contributors reject a spending request, the contract unlocks the es- crow and allows contributors to securely withdraw their investments.

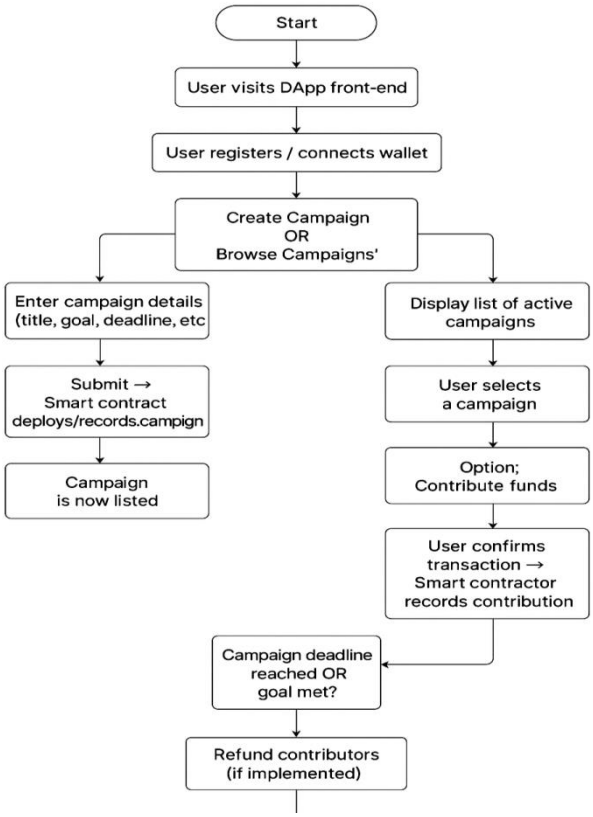


Fig2.System Workflow

V. Implementation Details

A. Development Environment and Technology Stack

The implementation of the decentralized application (dApp) requires a hybrid technology stack integrating modern web technologies with blockchain infrastructure.

- **Frontend:** The user interface is developed using React.js and Next.js to ensure a responsive, dynamic, and cross-platform user experience.
- **Backend:** Node.js is utilized for auxiliary server-side operations and routing where required.
- **Blockchain Development Environment:** Frameworks such as Hardhat and Truffle are employed for compiling, debugging, testing, and deploying smart contracts.
- **Programming Language:** Solidity is used for implementing smart contracts on the Ethereum Virtual Machine (EVM).

B. Smart Contract Development

The core business logic of the crowdfunding platform is implemented in Solidity, a high-level object-oriented programming language designed specifically for blockchain execution.

The smart contracts manage critical state variables including minimum contribution thresholds, campaign deadlines, escrow balances, contributor mappings, and voting records. In certain implementations, standard token interfaces such as ERC-20 may be incorporated to issue proportional digital tokens representing contributor stakes.

C. Web3 Integration and Wallet Connectivity

To bridge the traditional web interface with the decentral-

ized blockchain network, Web3 integration libraries such as Web3.js or Ethers.js are used.

MetaMask serves as the primary authentication gateway, securely managing users' private keys locally within the browser. Users digitally sign transactions and authorize fund transfers without exposing sensitive credentials to centralized servers.

Remote Procedure Call (RPC) providers such as Infura or Alchemy are integrated to broadcast signed transactions to the distributed ledger. This approach eliminates the necessity for users to operate a full blockchain node while maintaining seamless network connectivity.

D. Off-Chain Data Storage

Storing large multimedia assets directly on-chain is econom- ically inefficient due to high gas costs and network congestion. Therefore, the system adopts a hybrid storage model.

Extensive campaign metadata—including images, videos, and detailed textual descriptions—is stored on the InterPlan- etary File System (IPFS). IPFS generates a unique cryp- tographic Content Identifier (CID) for each file. Only this lightweight hash is stored within the Solidity smart contract, ensuring data integrity, immutability, and cost efficiency.

E. Testing and Testnet Deployment

Before mainnet deployment, the smart contracts undergo rigorous testing and simulation.

Automated unit testing is performed using the Mocha test- ing framework to verify functional correctness under various edge-case scenarios, including deposit handling, milestone approvals, and refund logic.

The decentralized application is subsequently deployed to public test networks such as Goerli, Sepolia, or the Polygon Mumbai testnet. These environments allow developers to use simulated cryptocurrency to evaluate gas consumption, validate transaction flows, and ensure full end-to-end system stability before final deployment to the Ethereum mainnet.

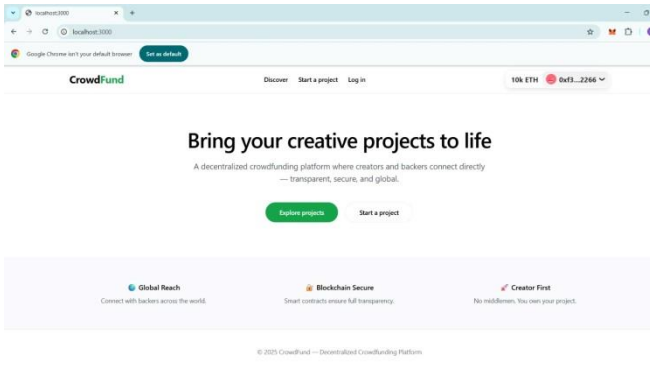


Fig3.web Interference

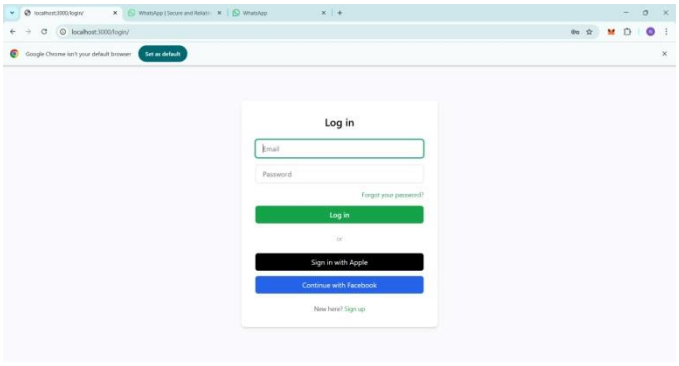


Fig4. Login credential

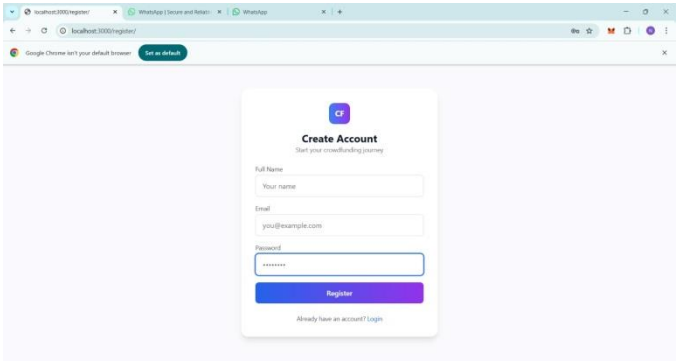


fig5. Create Account

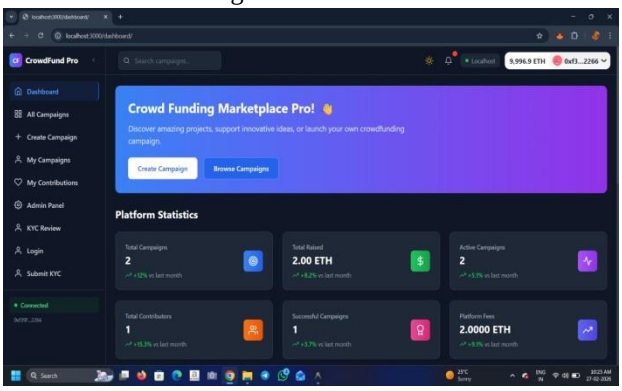
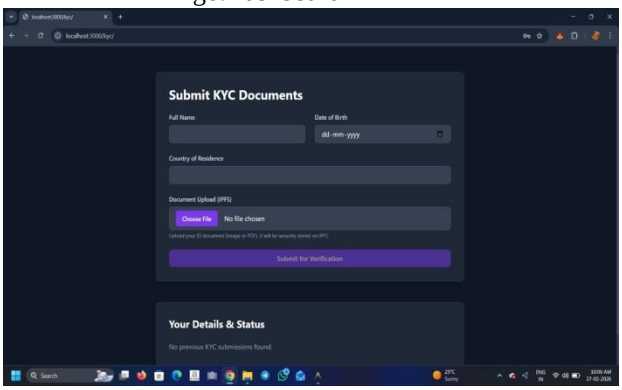


Fig6.Dashboard



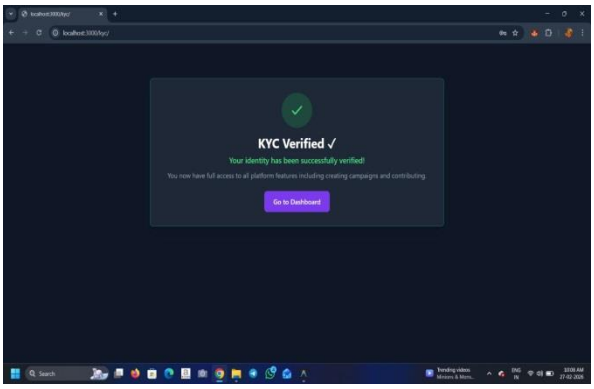


Fig.7 KYC Verification

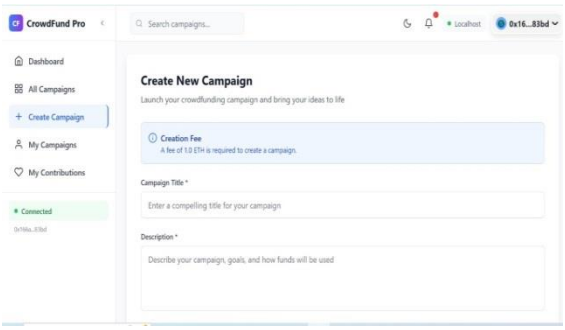


Fig8.Create Campaign

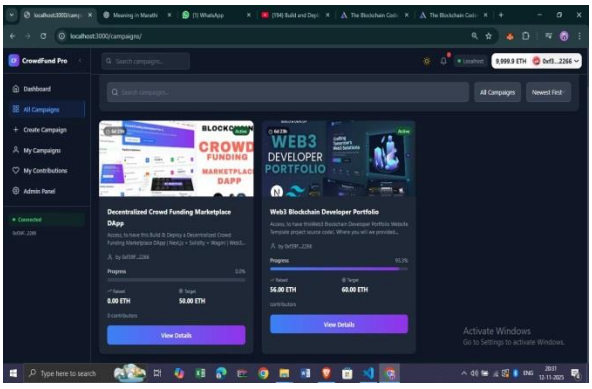
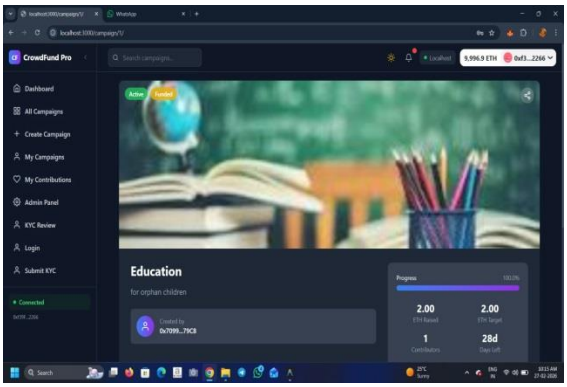


Fig9.Campaign History



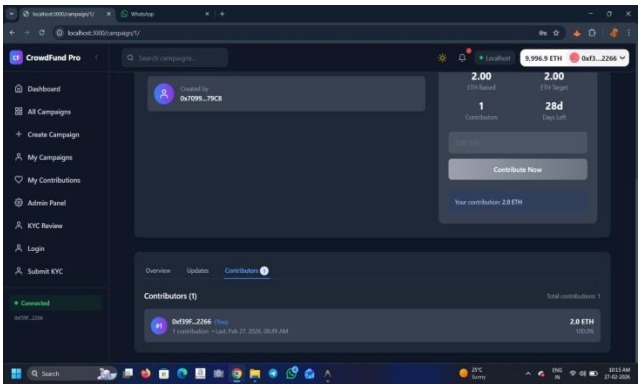


Fig 10.Transcation Detail

VI. Performance Evaluation

Benchmarking results indicate:

- Layer-1 Ethereum Gas Cost: $1.78 per transaction
- Layer-2 Polygon Gas Cost: ¡ $0.01
- Increased transaction throughput
- Improved micro-investment feasibility

VII. Security Analysis

The system mitigates major vulnerabilities:

- Reentrancy Attacks
- Integer Overflow / Underflow
- Access Control Exploits

Security validation includes formal verification tools and unit testing frameworks.

VIII. Conclusion and Future Work

A. Conclusion

This study presented the design and implementation of a blockchain-based crowdfunding platform using Ethereum smart contracts. By replacing centralized intermediaries with a decentralized escrow mechanism, the system improves transparency, reduces administrative costs, and minimizes fraud risks. The integration of Layer-2 scaling solutions enhances transaction efficiency by lowering gas costs and increas- ing throughput. Additionally, security auditing mechanisms strengthen protection against vulnerabilities such as reentrancy and integer overflow attacks. Overall, the proposed architecture offers a secure, transparent, and scalable model for decentralized crowdfunding.

B. Future Research Directions

Future work can focus on:

- **AI Integration:** Applying machine learning to assess campaign viability and detect fraud.
- **Cross-Chain Interoperability:** Enabling funding across multiple blockchain networks.
- **Regulatory Compliance and Privacy:** Incorporating privacy-preserving techniques such as zero-knowledge proofs while aligning with evolving regulations.

ACKNOWLEDGMENT

The authors, Amoli Dhuri, Nikita Jarad, and Nikita Mane, would like to express their sincere gratitude to Usha Mittal Institute of Technology, SNDT Women's University, Mumbai, for providing the essential technical infrastructure, laboratory resources, and academic environment that facilitated this research.

We would also like to express our heartfelt thanks to our respected Principal, Dr.Yogesh Nerkar, for his constant encouragement, visionary leadership, and support in fostering an innovative academic atmosphere that promotes research and technical excellence.

We sincerely acknowledge the support and guidance of our Head of Department, Dr.Rachana Dhanawat, for her valuable suggestions, administrative support, and motivation throughout the course of this project.

Special thanks are extended to our project guide, Mr. Sanjaykumar Ranveer, whose invaluable mentorship, technical expertise, and continuous encouragement were instrumental in shaping the architectural design and security evaluation of the proposed blockchain-based crowdfunding system.

Furthermore, we acknowledge the contributions of the Ethereum Foundation and the broader Web3 developer community. Their extensive documentation, development frameworks, and open-source resources significantly supported the smart contract implementation and optimization phases of this project. We also thank our peers and colleagues for their constructive feedback and valuable discussions, which enhanced the overall quality and rigor of this work.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] V. Buterin, "A next-generation smart contract and decentralized ap- plication platform," Ethereum White Paper, 2014. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[3] P. B. Kumbharkar, G. Asati, R. Palaskar, A. Valekar, and S. Yenegure, "Fund Crypt: Blockchain based crowdfunding platform using SHA- 256 & PoS algorithm," in *Proc. 2nd Int. Conf. Edge Comput. Appl. (ICECAA)*, Pune, India, 2023, pp. 316–323.

[4] A. K. Dubey, S. C. Shingte, M. S. Siddiqui, and S. Patil, "Crowdfunding using blockchain for startups and investors," in *Proc. 7th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Vasai, India, 2023, pp. 1400–1405.

[5] A. Kumar, S. Utekar, and V. D, "Crowdfunding platform using decen- tralised application (blockchain)," in *Proc. Int. Conf. Commun. Comput. Sci. Eng. (IC3SE)*, Kattankulathur, India, 2024, pp. 412–417.

[6] K. Mukherjee, A. Rana, and S. Rani, "Crowdfunding platform using blockchain," in *Proc. IEEE 9th Int. Conf. Converg. Technol. (I2CT)*, Pune, India, 2024, pp. 1–6.

[7] H. S. Rao, P. Sinha, V. P. Aniketh, S. Subbaiah, and M. Namratha, "Blockchain based crowdfunding platforms: Exploratory literature sur- vey," in *Proc. 5th Biennial Int. Conf. Nascent Technol. Eng. (ICNTE)*, Bengaluru, India, 2023, pp. 1–4.

[8] R. N. Kumaran, S. Geetha, S. Kaushik, C. Kishore, and A. N. Rathish, "Blockchain based crowd funding," in *Proc. Int. Conf. Comput. Com- mun. Inform. (ICCCI)*, Coimbatore, India, 2023, pp. 1–8.

[9] M. Zichichi, M. Contu, S. Ferretti, and G. D'Angelo, "LikeStarter: A smart-contract based social DAO for crowdfunding," in *IEEE INFOCOM Workshops*, Paris, France, 2019, pp. 313–318.

[10] V. Hassija, V. Chamola, and S. Zeadally, "BitFund: A blockchain- based crowd funding platform for future smart and connected nation," *Sustainable Cities and Society*, vol. 60, p. 102145, 2020.

[11] E. Mollick, "The dynamics of crowdfunding: An exploratory study," *Journal of Business Venturing*, vol. 29, no. 1, pp. 1–16, 2014.

[12] M. N. Saadat, S. A. Halim, H. Osman, R. M. Nassr, and M. F. Zuhairi, "Blockchain based crowdfunding systems," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 15, no. 1, pp. 409–413, 2019.