

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANASANGAMA, BELAGAVI - 590018



Project Work Phase II Report (21AML183)

On

CROWDFUNDING PLATFORM USING BLOCKCHAIN

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Submitted by

1BG21AI073 NAIRA ANJUM

1BG21AI082 PRANAMYA R BAIRY

1BG21AI101 SHIVEN DASHORA

Under the guidance of

Prof. Divya M S

Associate professor, Dept. of AIML, BNMIT, Bengaluru



B.N.M. Institute of Technology

An Autonomous Institution under VTU

Approved by AICTE, Accredited as grade A Institution by NAAC. All eligible branches – CSE, ECE, EEE, ISE & Mech. Engg. are Accredited by NBA for academic years 2018-19 to 2024-25 & valid upto 30.06.2025. URL: www.bnmit.org

Department of Artificial Intelligence and Machine Learning

2024 – 2025

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Approved by AICTE, Accredited as grade A Institution by NAAC. All eligible branches – CSE, ECE, EEE, ISE & Mech. Engg. are Accredited by NBA for academic years 2018-19 to 2024-25 & valid upto 30.06.2025. URL: www.bnmit.org

Department of Artificial Intelligence and Machine Learning



Vidyayāmruthamashnuthe

CERTIFICATE

This is to certify that the project work phase – II titled **Crowdfunding Platform Using Blockchain** carried out by **Naira Anjum (1BG21AI073)**, **Pranamy R Bairy (1BG21AI082)**, **Shiven Dashora (1BG21AI101)** the bonafide students of VIII Semester B.E., **B.N.M Institute of Technology**, in partial fulfillment for the award of degree in **Bachelor of Engineering** in **ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING** of the **Visvesvaraya Technological University**, Belagavi during the year 2024-25. This report has been approved as it satisfies the academic requirements in respect of Project Work Phase – II (21AML183) prescribed for the said degree.

Prof. Divya M S

Assistant professor,
Dept. of AIML
BNMIT, Bengaluru

Dr. Sheba Selvam

Professor and HOD
Dept. of AIML
BNMIT, Bengaluru

Dr. S.Y. Kulkarni

Additional Director
and Principal
BNMIT, Bengaluru

Name

Signature

Examiner 1:

Examiner 2:

DECLARATION

We, Naira Anjum (1BG21AI073), Pranamya R Bairy (1BG21AI082), Shiven Dashora (1BG21AI101) hereby declare that the Project Work Phase-II (21AML183) entitled “Crowdfunding Platform Using Blockchain” has been independently carried out by us under the guidance of Prof. Divya M S, Assistant professor, department of AIML, BNM Institute of Technology, Bengaluru, in partial fulfillment of the requirements of the degree of Bachelor of Engineering in Artificial Intelligence & Machine Learning of the Visvesvaraya Technological University, Belagavi.

We further declare that we have not submitted this report either in part or in full to any other university for the reward of any degree.

Place: Bengaluru

Date:

Naira Anjum

Pranamya R Bairy

Shiven Dashora

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned my effort with success.

We would like to thank **Shri Narayan Rao R. Maanay**, Secretary, BNMEI, Bengaluru for providing an excellent academic environment in the College.

We would like to sincerely thank **Prof. T. J. Rama Murthy**, Director, BNMIT, Bengaluru, for having extended his support and encouraging me during the course of the work.

We would like to sincerely thank **Dr. S.Y. Kulkarni**, Additional Director and Principal BNMIT, Bengaluru, for having extended his support and encouraging me during the course of the work.

We would like to express my gratitude to **Prof. Eishwar N. Maanay**, Dean, BNMIT, Bengaluru for his relentless support, guidance and assistance.

We would like to thank **Dr. Krishnamurthy G.N**, Deputy Director, BNMIT, Bengaluru for his constant encouragement.

We would like to thank **Dr. Sheba Selvam**, Professor and Head of the Department of Artificial Intelligence and Machine Learning, BNMIT, Bengaluru who has shared her opinions and thoughts which helped me in completion of my project successfully.

We would also like to thank my Guide **Prof. Divya M S**, Assistant professor, Department of Artificial Intelligence and Machine Learning, BNMIT, Bengaluru for guiding in a systematic manner.

Finally, We would like to thank all technical and non-technical faculty members of Department of Artificial Intelligence and Machine Learning, BNMIT, Bengaluru, for their support. We would like to thank our Family and Friends for their unfailing moral support and encouragement.

Naira Anjum (1BG21AI073)
Pranamy R Bairy (1BG21AI082)
Shiven Dashora (1BG21AI101)

ABSTRACT

Crowdfunding has become a crucial method for individuals and organizations to raise capital by pooling contributions from a broad audience. However, traditional crowdfunding platforms often encounter significant challenges, including high transaction fees, lack of transparency, and trust issues, which can hinder their effectiveness. This paper delves into the development of a blockchain-powered crowdfunding platform designed to address these challenges. By leveraging the decentralized, immutable, and transparent nature of blockchain technology, the platform ensures secure and efficient fund management, enhances trust among stakeholders, and minimizes reliance on intermediaries. The integration of smart contracts automates fund disbursement based on predefined conditions, fostering greater accountability and significantly reducing the risk of fraud. Key features such as real-time tracking of contributions, tamper-proof records, and tokenization for rewards or equity enhance user engagement and create a more robust crowdfunding experience. Additionally, this paper evaluates the scalability, security, and privacy aspects of the platform while identifying potential challenges and proposing viable solutions. By demonstrating how blockchain technology can revolutionize crowdfunding, this study highlights its potential to foster inclusivity, improve transparency, and establish a more equitable financial ecosystem. Through a comprehensive analysis, we showcase how this innovative approach can empower creators, backers, and businesses alike, ultimately reshaping the future of crowdfunding.

List of Figures

Sl. No.	Name	Page No.
4.2.1	System Architecture Design	23
4.3.1.1	Data Flow Diagram (Level 0)	25
4.3.2.1	Data Flow Diagram (Level 1)	26
4.4.1	Activity Diagram	27
4.5.1	Use Case Diagram	28
4.6.1	Sequence Diagram	31
5.5.1.1	Implementation of Frontend Pseudocode	36
5.5.2.1	Backend Pseudocode	39
6.1.1	Campaign Creation	42
6.1.2	All Campaigns	42
6.1.3	Campaign Details	42
6.1.4	Donating To Campaign	43
6.1.5	Transaction Request	43
6.1.6	Donating	44
6.1.7	Create Request By Campaign Manager	44
6.1.8	Voting to the Campaign	44
6.1.9	Funds Released	45
6.1.10	Details of the Request	45
6.1.11	Recorded in Etherscan	45

List of Tables

Sl. No.	Name	Page No.
2.1	Summary of Literature Survey	14

Table of Contents

Chapter No.	Title	Page No.
	DECLARATION	i
	ACKNOWLEDGEMENT	ii
	ABSTRACT	iii
	List of Figures	iv
	List of Tables	v
1	Introduction	1
	1.1 Motivation	1
	1.2 Problem Statement	2
	1.3 Objectives	2
	1.4 Summary	2
2	Literature Survey	3
	2.1 Introduction	3
	2.2 Literature Survey	3
	2.3 Summary	17
3	System Requirement Specifications	18
	3.1 Hardware Requirements	18
	3.2 Software Requirements	19
	3.3 Functional Requirements	19
	3.4 Non-Functional Requirements	20
	3.5 Summary	20
4	System Design	21
	4.1 High Level Design	21
	4.2 System Architecture	23
	4.3 Data Flow Diagram	24
	4.3.1 Data Flow Diagram (Level 0)	25
	4.3.2 Data Flow Diagram (Level 1)	26
	4.4 Activity Diagram	27
	4.5 Use-Case Diagram	28
	4.6 Sequence Diagram	31
	4.7 Summary	32
5	Implementation	33
	5.1 Introduction	33
	5.2 Programming Language Selection	33
	5.3 Platform/Framework	34
	5.4 Tools	35
	5.5 Implementation	35
	5.6 Summary	40
6	Results and Discussion	41
	6.1 Results	41
	6.3 Summary	46
7	Conclusion and Future Scope	47
	7.1 Conclusion	47
	7.2 Future Scope	47
		49

Appendix – I

Sustainable Development Goals

49

Appendix – II

Paper Publication Details

References

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

The advent of blockchain technology has revolutionized various industries, including crowdfunding, by addressing traditional challenges such as high operational costs, lack of transparency, and potential fraud. Blockchain-powered crowdfunding platforms leverage decentralization, immutability, and transparency to create a more secure and efficient funding model. By incorporating smart contracts, these platforms ensure that fund disbursements adhere to predefined conditions, fostering greater trust and accountability. Additionally, the removal of intermediaries reduces transaction costs and enhances efficiency, making the funding process more seamless and reliable.

Blockchain-based crowdfunding transcends geographical limitations, enabling creators and contributors worldwide to connect effortlessly. Contributors benefit from unparalleled transparency, allowing them to track their funds and ensure projects adhere to agreed terms. This model is particularly advantageous for small and medium-sized enterprises, startups, and philanthropic initiatives that struggle with traditional funding barriers. By fostering a secure, inclusive, and transparent ecosystem, blockchain-powered crowdfunding has the potential to reshape the funding landscape, creating equitable opportunities for creators and backers alike while addressing long-standing inefficiencies in resource allocation.

1.1 Motivation

The growing need for efficient and transparent financial systems has propelled the adoption of blockchain technology across various sectors, including crowdfunding. Traditional crowdfunding platforms often face issues like high transaction fees, fraud, and limited transparency, which can undermine trust between creators and contributors. These challenges highlight the need for a solution that fosters greater accountability and inclusiveness. Blockchain technology offers the ideal foundation to overcome these barriers by leveraging its decentralized and immutable nature. Another driving factor is the globalization of funding opportunities. Many creators and contributors from remote or underrepresented regions are excluded from traditional platforms due to geographical and financial barriers. Blockchain seamlessly promotes financial inclusion and equitable access to resources.

1.2 Problem Statement

Traditional crowdfunding platforms are hindered by challenges such as high transaction fees, limited transparency, susceptibility to fraud, and geographical restrictions. These issues reduce trust and participation, especially for small-scale creators and contributors from underprivileged regions. There is a need for an innovative solution that eliminates intermediaries, ensures transparent fund management, and fosters global accessibility for equitable participation in funding opportunities.

1.3 Objectives

- To develop a decentralized crowdfunding platform using blockchain technology
- To implement smart contracts
- To enable global accessibility and inclusivity
- To integrate tokenization mechanisms
- To ensure low transaction costs
- To promote transparency and accountability

1.4 Summary

This project aims to revolutionize the crowdfunding ecosystem by developing a blockchain powered platform that overcomes the common challenges faced by traditional systems, such as high transaction fees, limited transparency, and the risk of fraud. Leveraging blockchain's decentralization, the platform ensures security and transparency by recording every transaction on an immutable ledger. Smart contracts automate the funding process, ensuring that contributions are only released when predefined conditions are met, thereby building trust and eliminating the need for intermediaries. Tokenization further enhances engagement by offering digital tokens as rewards or incentives for contributors, which can hold intrinsic value or be traded on secondary markets. The platform also promotes global accessibility, allowing creators from all regions to showcase their ideas and connect with a diverse pool of backers. crowdfunding process, driving innovation and social impact. Additionally, by reducing operational costs and increasing efficiency, the platform offers a more cost-effective and scalable solution compared to traditional crowdfunding models.

CHAPTER 2

LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

Crowdfunding has evolved as an essential mechanism for entrepreneurs, non-profits, and individuals seeking financial support for their ideas, projects, and ventures. Traditional crowdfunding platforms, however, often encounter issues related to high fees, lack of transparency, and limited accessibility for a global audience. Blockchain technology presents an innovative solution by addressing these issues, offering a decentralized platform for secure, transparent, and cost-effective fundraising. By leveraging blockchain's immutability, smart contracts, and tokenization, this project aims to create a more efficient crowdfunding model, enhancing the overall user experience and expanding access to fundraising opportunities.

2.2 Literature Survey

This study [1] presents a comprehensive overview of the development and potential of blockchain-powered crowdfunding platforms. It examines how blockchain technologies such as smart contracts, decentralized ledgers, and tokenization are transforming traditional crowdfunding models by enhancing transparency, reducing reliance on intermediaries, and improving fund security. The authors discuss the implementation of key components like Ethereum-based smart contracts and decentralized storage solutions to ensure trust and accountability throughout the funding process.

Challenges discussed in the study [1] include the scalability of blockchain networks under high transaction volumes, the complexity of smart contract development, and the integration of blockchain with user-friendly interfaces. The paper also highlights limitations related to regulatory concerns, energy consumption, and adoption barriers among non-technical users. By consolidating these findings, the study [1] outlines the current landscape and serves as a guide for future innovations aimed at creating more accessible, secure, and efficient crowdfunding ecosystems through blockchain technology.

The study [2] introduces an innovative blockchain-based approach to enhance the security and efficiency of crowdfunding platforms. By integrating smart contracts with decentralized storage systems and token-based incentives, the proposed solution ensures automatic fund disbursement, transparent tracking of contributions, and improved trust between creators and backers. The system is designed to eliminate intermediaries and reduce transaction costs while offering real-time visibility into project progress.

While the approach demonstrates potential in creating a more reliable and user-centric crowdfunding model, the authors acknowledge certain limitations. These include the complexity of smart contract deployment, scalability issues under high user loads, and challenges in ensuring regulatory compliance across jurisdictions. Additionally, non-technical users may face barriers in interacting with blockchain interfaces. Nonetheless, this study [2] marks a significant step toward integrating blockchain with financial technologies to build more secure, inclusive, and transparent funding ecosystems. The study [3] offers a detailed analysis of blockchain integration in crowdfunding platforms, focusing on how decentralized technologies can reshape the funding landscape. It explores the role of smart contracts in automating fund disbursement and ensuring compliance with predefined rules, while decentralized storage systems enhance transparency and security. Tokenization mechanisms are also discussed as tools for incentivizing participation and representing ownership or rewards. Together, these components contribute to a more secure, efficient, and trust-driven crowdfunding process. Despite these advancements, the study [3] notes several challenges that need to be addressed for wider adoption. These include the steep learning curve for developers and users unfamiliar with blockchain, difficulties in managing regulatory and legal uncertainties, and concerns over scalability and energy consumption in public blockchain networks. The authors emphasize the need for more user-friendly interfaces, scalable infrastructures, and adaptable regulatory frameworks to fully unlock the potential of blockchain in crowdfunding.

The study [4] explores the integration of decentralized identity verification systems within blockchain-based crowdfunding platforms. By using decentralized identifiers (DIDs) and verifiable credentials, the system ensures that only verified users can participate, thus minimizing the risk of fraud and unauthorized access. This enhances trust and transparency across the platform, which is essential for attracting both

backers and project creators. The study also discusses how these systems can work without compromising user privacy, thanks to blockchain's inherent security features. However, challenges remain in terms of interoperability with existing identity systems and adherence to global data protection regulations like GDPR.

The study [5] proposes the use of Layer 2 blockchain solutions, such as sidechains and rollups, combined with optimized smart contracts to improve the scalability and cost-efficiency of crowdfunding platforms. These enhancements allow the system to handle more transactions at a lower cost, making micro-contributions and frequent transactions more practical. The study shows that this approach significantly reduces network congestion and gas fees, which are common issues in traditional blockchain applications. However, it also highlights synchronization delays between Layer 1 and Layer 2 networks as a potential bottleneck, suggesting future work in refining contract design and ensuring seamless fund settlement across layers.

The study [6] delves into the socio-economic implications of blockchain-powered crowdfunding, emphasizing its role in democratizing access to capital for marginalized communities, grassroots innovators, and nonprofit initiatives. By removing intermediaries and offering global access, blockchain allows projects that may have been overlooked by traditional funding systems to gain traction. It also highlights the transparency and traceability features of blockchain as tools for fostering accountability in socially driven projects. Nevertheless, the study points out that widespread adoption is hindered by limited blockchain literacy and technical barriers. To overcome this, the authors recommend the development of intuitive interfaces, community engagement strategies, and educational outreach to bridge the digital divide and promote equitable participation.

The study [6] highlights the societal and functional benefits of blockchain-powered crowdfunding platforms, especially in promoting financial inclusivity and transparency. By leveraging blockchain's decentralized architecture, these platforms enable secure, traceable, and immutable transactions, empowering small-scale entrepreneurs, NGOs, and underrepresented innovators to access funding beyond traditional financial systems. The study emphasizes how such platforms can foster

trust among contributors by providing real-time access to fund flow and enforcing smart contract-based conditions for fund disbursement. However, it also raises concerns about the digital divide, pointing out that a lack of blockchain awareness and accessibility may hinder adoption in low-tech communities. The authors advocate for user-friendly interfaces and educational initiatives to increase participation and ensure equitable access.

The study [7] provides a technical evaluation of decentralized crowdfunding systems, focusing on smart contract architecture, tokenization mechanisms, and integration with identity verification protocols. It explores how token-based models can incentivize contributions through reward systems, equity sharing, or governance participation. Additionally, the study outlines the challenges of ensuring regulatory compliance across borders, particularly in jurisdictions with strict crowdfunding or data protection laws. The need for scalable, interoperable, and secure contract logic is also emphasized, as poorly designed contracts may lead to vulnerabilities or misuse. The authors suggest the development of standardized frameworks for smart contract deployment and cross-chain functionality to enhance platform security and reliability, ultimately promoting wider adoption and operational efficiency.

The study [8] explores the development of a decentralized crowdfunding platform using blockchain technology to enhance transparency, trust, and automation in fundraising processes. By integrating smart contracts, the system enables automatic fund disbursement based on predefined conditions, minimizing the risk of fraud or misuse. Blockchain's immutable ledger ensures that all transactions and contribution records are tamper-proof and verifiable,

addressing major concerns in traditional crowdfunding models. However, the study points out challenges in integrating real-time data and user-friendly interfaces. While the backend ensures robust security and transparency, the frontend experience can be hindered by the complexity of interacting with blockchain elements. The authors suggest the inclusion of simplified interfaces and blockchain abstraction layers to improve accessibility for non-technical users.

The study [9] introduces a blockchain-based crowdfunding solution tailored for real-time project tracking and token-based rewards. The system employs blockchain to facilitate transparent fund management, allowing contributors to monitor project milestones and fund utilization in real-time. Tokenization is used to incentivize contributions by offering digital rewards or equity, encouraging long-term engagement. Despite these innovations, the study highlights issues related to scalability and network latency, especially during high-volume transactions. Additionally, legal and regulatory uncertainties surrounding tokenized fundraising present obstacles for global deployment. The authors recommend further exploration into scalable blockchain networks and adaptable legal frameworks to ensure the platform's effectiveness and compliance across different regions.

The study [10] presents a blockchain-based crowdfunding platform that leverages smart contracts to manage the funding lifecycle, from project creation to fund disbursement. The system emphasizes transparency and reliability, using blockchain's immutable nature to store records of transactions and contributor details. Smart contracts automate processes like milestone-based fund release, reducing the potential for human error or manipulation. However, the study identifies that the platform's performance heavily depends on the usability and clarity of the frontend interface. In particular, non-technical users may struggle with understanding wallet interactions and transaction confirmations. The authors recommend incorporating more intuitive UI/UX design and providing educational prompts to enhance user engagement.

The study [11] introduces an advanced architecture integrating blockchain with a generative model to simulate various funding scenarios. This approach allows project creators to visualize and optimize funding strategies before launching live campaigns. By simulating different contribution patterns and risk factors, the platform aims to improve campaign planning and donor trust. However, the integration of such models increases system complexity and requires significant computing power, which can slow down performance on standard devices. The study also notes that without sufficient real-world data, simulations may not accurately predict user behavior. Future improvements may include integrating real-time analytics and feedback systems to continuously refine simulation accuracy and campaign performance.

The study [12] explores a decentralized crowdfunding model that introduces a priority-based attention mechanism for fund distribution. The system focuses on distributing funds based on project urgency or popularity, ensuring timely support for high-impact initiatives. This mechanism is particularly useful for platforms dealing with multiple ongoing campaigns, helping users decide where to contribute. However, the approach can struggle in scenarios with incomplete project metadata or when dealing with diverse funding goals across different regions. The authors propose enhancing adaptability by including metadata standardization and multilingual support to make the platform suitable for broader and more inclusive use cases.

The study [13] presents an innovative system that combines Convolutional Neural Networks (CNNs), Bi-directional Recurrent Neural Networks (RNNs), and Connectionist Temporal Classification (CTC) for converting handwritten text into digital form. In this hybrid architecture, CNNs are utilized for feature extraction from handwritten input images, capturing key spatial information such as character shapes and strokes. The Bi-directional RNNs are responsible for sequence modeling, allowing the system to understand the context by analyzing both forward and backward dependencies in the text.

Finally, CTC is employed to align the extracted features with the target transcription without requiring pre-segmented input, enabling end-to-end training. This method demonstrates promising accuracy, especially for simple scripts and well-formed handwriting. However, a key limitation lies in its reliance on large and diverse training datasets, which are not always readily available—particularly for rare or non-standard handwriting styles. The system's accuracy also decreases when dealing with complex layouts or irregular handwriting formats. The authors of study [13] recommend enhancing generalization capabilities to broaden the applicability of the model across various writing conditions and scripts.

The study [14] introduces a machine learning-based approach for handwritten character recognition by integrating CNNs and Support Vector Machines (SVMs). CNNs are used to extract relevant features from the input images, while SVMs serve as the classification module to recognize individual characters. This approach proves effective in controlled environments with clean, standardized handwritten inputs. The system achieves a high recognition rate in scenarios with well-formed and isolated characters. However, it struggles with complex or messy handwriting, particularly when characters are connected or written in cursive styles. One of the key limitations highlighted is the system's reduced robustness in real-world applications where handwriting variations are common. Additionally, performance heavily depends on the quality and quantity of labeled training data. To address these challenges, the study [14] proposes the incorporation of data augmentation techniques and transfer learning strategies to improve generalization and adaptability to a wider array of handwriting conditions.

The study [15] proposes a deep learning-based handwritten text recognition model that integrates CNNs, RNNs, and CTC loss, implemented using the TensorFlow framework. This system is designed to handle end-to-end recognition tasks, especially those involving short text sequences such as individual words or short phrases. CNNs are responsible for extracting features, RNNs handle the sequence prediction, and CTC loss facilitates the alignment between input and output without explicit character-level annotation. The approach yields satisfactory results when dealing with clear, segmented text.

However, its performance drops significantly when processing longer sentences or cursive scripts, where the fluid connections between letters introduce ambiguity in recognition. The sequential dependencies become harder to model accurately, leading to increased recognition errors. Study [15] emphasizes the need for further optimization, especially in training the system to manage longer sequences and more complex writing styles. It also suggests that improvements in model architecture and the use of more diverse datasets could substantially enhance recognition accuracy and system reliability.

The study [15] introduces a decentralized crowdfunding platform built on Ethereum blockchain, utilizing smart contracts to automate the fundraising process and ensure transparency. The platform allows project creators to launch campaigns, receive contributions, and distribute rewards without relying on centralized intermediaries. The use of blockchain ensures immutability and traceability of all transactions, enhancing donor trust. However, the study [15] notes that high transaction fees (gas fees) and network congestion can hinder usability, especially for micro-donations. The authors recommend exploring layer-2 scaling solutions and alternative blockchains to reduce operational costs and improve efficiency.

The study [16] investigates the implementation of a token-based crowdfunding model, where backers receive utility tokens in exchange for their contributions. These tokens can represent future access to products, voting rights, or other utilities within the platform. The model promotes community engagement and incentivizes long-term participation. Despite its advantages, the study [16] highlights regulatory uncertainty around token issuance, which can pose legal risks for startups. It suggests incorporating compliance frameworks and Know Your Customer (KYC) mechanisms to align with global financial regulations while preserving decentralization.

The study [17] presents a hybrid crowdfunding model that integrates blockchain with traditional financial infrastructure. The proposed system leverages smart contracts for fund management but uses fiat on-ramps for broader accessibility. This approach helps bridge the gap between crypto-savvy users and conventional investors. Nevertheless, the study [17] points out challenges in interoperability between blockchain platforms and banking systems, which may delay fund transfers or introduce technical bottlenecks. The authors recommend developing standardized APIs and cross-platform protocols to streamline the integration process.

The study [18] explores the use of decentralized autonomous organizations (DAOs) in managing crowdfunding campaigns. By delegating campaign governance to token holders, the platform enhances transparency and democratizes decision-making. Donors can vote on how funds are allocated, increasing accountability and reducing misuse. However, the study [18] identifies potential drawbacks, such as voter apathy and the risk of manipulation by wealthy stakeholders. The authors suggest implementing reputation systems and quadratic voting mechanisms to ensure fair participation and better reflect community consensus.

The study [19] addresses the challenge of limited user participation and data scarcity in newly launched blockchain-based crowdfunding platforms. To overcome the initial lack of user interaction and trust, the authors propose an innovative approach leveraging adversarial learning techniques in decentralized identity and behavioral modeling. This involves training a generator to simulate realistic user behavior and campaign interactions, while a discriminator evaluates their authenticity—thereby enabling the creation of synthetic datasets that closely resemble actual user activity on the platform.

The study [19] addresses the challenge of limited user participation and data scarcity in newly launched blockchain-based crowdfunding platforms. To overcome the initial lack of user interaction and trust, the authors propose an innovative approach leveraging adversarial learning techniques in decentralized identity and behavioral modelling.

This involves training a generator to simulate realistic user behavior and campaign interactions, while a discriminator evaluates their authenticity—thereby enabling the creation of synthetic datasets that closely resemble actual user activity on the platform.

Such synthetic interaction data aids in stress-testing the platform, analyzing fraud detection mechanisms, and optimizing smart contract workflows before real-world deployment. This technique is especially useful in early-stage platforms, where real transaction data is sparse, and evaluating system performance under various scenarios is critical.

The study specifically focuses on underrepresented crowdfunding use cases, such as regional or grassroots community projects, which often struggle with visibility and data availability. One significant benefit of this adversarial learning approach is its ability to enhance the system's generalization, allowing it to adapt to a broad range of campaign types and user behaviors.

However, the authors also highlight challenges, including the computational cost associated with training generative models and the need for fine-tuned adversarial balancing to avoid generating unrealistic or biased synthetic data. The study [19] emphasizes the potential of adversarial learning not only to mitigate data scarcity but also to improve trust, scalability, and robustness in blockchain-based crowdfunding ecosystems.

Paper Name	Year	Approach with Description	Drawbacks
Decentralized Crowdfunding Using Blockchain and DAOs [1]	2024	Proposed a smart contract-based crowdfunding platform integrated with DAO mechanisms for fund governance.	Limited scalability due to high gas fees on Ethereum.
IPFS-Based Storage in Crowdfunding Applications [2]	2023	Integrated IPFS for decentralized proof storage linked with a crowdfunding dApp on the Polygon network.	Proof retrieval delays due to IPFS network latency
Smart Contracts for Transparent Fundraising [3]	2023	Designed Ethereum-based smart contracts to automate fund release based on pre-set milestones.	Relies heavily on manual proof validation.
Blockchain-Based Crowdfunding Platform [4]	2022	Demonstrated the use of MetaMask for secure user authentication and Web3.js for blockchain interaction.	Focused only on authentication; did not address scalability or DAO integration.
Decentralized Governance with DAOs.[5]	2022	Explored DAO governance models for community.	Complex voting mechanisms

Crowdfunding Through Blockchain: A Case Study.[6]	2021	Studied real-world crowdfunding platforms leveraging blockchain.	Lacked Detailed analysis of storage for proofs.
Scalability in Blockchain-Based Applications.[7]	2021	Proposed a layer-2 solution for reducing transaction costs and improving scalability in blockchain applications.	Not specifically tailored for crowdfunding scenarios.
Integration of IPFS with Smart Contracts.[8]	2020	Showcased the integration of IPFS with Ethereum smart contracts for data storage and reference.	High reliance on off-chain components for data availability.
Decentralized Identity Management Using MetaMask.[9]	2020	Discussed MetaMask for decentralized identity and transaction signing in crowdfunding systems.	Focused only on identity management without exploring DAO or proof-based fund release.
Transparency in Crowdfunding with Blockchain.[10]	2019	Proposed a blockchain model where campaign details and transactions are visible to all stakeholders.	Did not incorporate fund utilization tracking or proof submission mechanisms
Distributed File Systems for Blockchain Applications [11]	2019	Reviewed decentralized storage systems like IPFS and their compatibility with blockchain platforms.	Storage systems' performance degraded under high traffic scenarios.

DAO: The Future of Governance," B. Taylor et al., Elsevier [12]	2018	Introduced DAO frameworks for automating decision-making in decentralized ecosystems.	Lack of clarity on dispute resolution and governance conflicts.
Blockchain Crowdfunding Models," F. Yang, Springer [13]	2018	Analyzed different crowdfunding models using blockchain, focusing on campaign transparency and donor trust.	Did not address the role of decentralized storage in proof validation.
Ethereum Smart Contracts for Transparency .[14]	2017	Implemented smart contracts to ensure transparency and trust in crowdfunding campaigns.	Lacked advanced mechanisms for proof verification and fund release.
Data Integrity in Decentralized Applications .[15]	2017	Proposed techniques to ensure data integrity using blockchain-based hashing and decentralized storage systems.	Did not address user-friendly retrieval mechanisms for non-technical users.
Security Challenges in Blockchain Systems .[16]	2016	Investigated common security threats in blockchain systems, including smart contract vulnerabilities.	Focused more on theoretical threats without real-world implementation examples.

Decentralized Storage Networks.[17]	2016	Discussed the design and implementation of decentralized storage systems for scalability and efficiency.	Limited real-world use cases in integration with blockchain applications.
Smart Contracts: Applications and Challenges.[18]	2015	Explored the applications of smart contracts in decentralized systems and discussed their technical challenges.	Did not cover DAO integration or crowdfunding-specific use cases.
Blockchain for Transparency and Security.[19]	2014	Introduced blockchain as a tool for enhancing transparency and security in financial applications.	Very general; lacked focus on crowdfunding platforms or use cases.
Introduction to Blockchain Technology.[20]	2013	Foundational research on blockchain's architecture, immutability, and decentralized nature.	Did not address smart contracts, DAOs, or application-specific frameworks.

Table 2.1: Summary of Literature Survey

2.3 Summary

Several blockchain-based crowdfunding platforms have been developed to test the viability and potential of blockchain in this space. For instance, platforms like FundRequest and Giveth have adopted blockchain to create decentralized fundraising systems. These platforms aim to minimize fees, increase transparency, and build trust through the use of blockchain's immutable ledger and smart contracts. Research into these platforms has shown promising results, indicating that blockchain can significantly reduce the risks of fraud and mismanagement in

crowdfunding campaigns. While blockchain-based crowdfunding is still in its early stages, its potential is immense. Decentralized Autonomous Organizations (DAOs) can empower communities to lead fundraising initiatives and automate fund distribution through smart contracts. Tokenized crowdfunding can democratize investment opportunities by allowing smaller contributions and global participation. Additionally, blockchain enables micropayments and fractional ownership, making crowdfunding accessible to a wider range of individual.

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATIONS

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATIONS

This chapter provides an overview of the system requirements necessary for the development and deployment of the Blockchain-Powered Crowdfunding Platform. It outlines the hardware and software prerequisites, along with the functional and non-functional requirements that ensure the system's optimal performance.

3.1 Hardware Requirements

The system requires specific hardware configurations for efficient blockchain integration, smart contract execution, and secure transactions. The recommended hardware specifications are:

Development Machine

- **Processor:** Intel Core i5/i7 or AMD Ryzen 5/7 (or higher)
- **RAM:** Minimum 16GB (32GB recommended for handling multiple blockchain nodes)
- **Storage:** At least 100GB free space (preferably SSD for fast transaction processing)
- **GPU:** Optional (recommended for cryptographic computations and scalability testing)
- **Operating System:** Windows 10/11, Ubuntu 20.04+, or macOS

Deployment Environment

- **Processor:** Any modern CPU (Intel Core i5/i7 or equivalent)
- **RAM:** 8GB minimum (16GB recommended for scalability)
- **Storage:** 50GB free space (SSD recommended)
- **GPU:** Not required for basic transactions (recommended for AI-enhanced analytics)
- **Server Hosting:** Cloud-based (AWS, Azure, or on-premises Linux server)

3.2 Software Requirements

The following software components are required for the development and execution of the project:

Development Environment

- **Programming Language:** Python 3.8+
- **IDE:** Jupyter Notebook, VS Code, or PyCharm
- **Libraries & Dependencies:**
 - TensorFlow/Keras (for model training and inference)
 - OpenCV (for image processing)
 - NumPy (for numerical computations)
 - Gradio (for building a user-friendly interface)

Deployment & Execution

- **Server Framework:** Flask or FastAPI (alternative to Gradio for web deployment)
- **Virtual Environment:** Anaconda or venv for dependency management

3.3 Functional Requirements

The functional requirements define the core operations that the system must perform:

- **Image Preprocessing:** Convert input images to grayscale, resize, normalize, and prepare for model input.
- **Model Loading:** Load the pre-trained CRNN model for inference.
- **Handwritten Text Recognition:** Process the input image and predict the corresponding text.
- **User Interaction:** Provide an interface for users to upload images and retrieve recognized text.

- **Error Handling:** Display error messages when invalid or corrupted images are uploaded.

3.4 Non-Functional Requirements

The non-functional requirements focus on performance, security, and usability aspects of the system:

- **Security:** Implement encryption and multi-signature wallets for secure transactions.
- **Scalability:** Support thousands of transactions per second with efficient blockchain consensus mechanisms.
- **Transparency:** Ensure all transactions are recorded on the blockchain and publicly auditable.
- **Performance:** Optimize smart contracts to minimize gas fees and execution time.
- **Usability:** Provide an intuitive and accessible user interface for seamless interaction.

3.5 Summary

The Blockchain-Powered Crowdfunding Platform ensures secure and transparent funding through smart contracts, decentralized storage, and real-time tracking. It requires a high-performance development machine and cloud-based deployment for scalability. The software stack includes Solidity, Node.js/Python, React.js, Ethereum/Hyperledger Fabric, Web3.js, and IPFS. Key features include user authentication, project creation, automated fund disbursement, and tokenization, with a focus on security, scalability, and usability. This platform eliminates intermediaries, enhancing trust and efficiency in crowdfunding.

CHAPTER 4

SYSTEM DESIGN

CHAPTER 4

SYSTEM DESIGN

The system design of the blockchain-based crowdfunding platform consists of a user-friendly frontend built with technologies like React, allowing users to create and browse campaigns, connect wallets, and contribute funds. The backend, developed using Node.js or Express (if used), manages non-sensitive metadata and interacts with the blockchain via Web3.js or Ethers.js. Wallet integration through MetaMask or similar tools enables secure user authentication and transactions. Additionally, decentralized storage like IPFS can be used to host campaign media and ensure data integrity in a tamper-proof manner.

4.1 High End Design

1. Frontend (User Interface Layer)

- **Technology Stack:** React.js, HTML5, CSS3, Web3.js, and MetaMask Integration
- **Functionality:**
 - User registration and authentication via crypto wallet (e.g., MetaMask)
 - Dashboard to create, browse, and contribute to crowdfunding campaigns
 - Display of real-time campaign progress (funds raised, contributors, milestones)
 - Integration with blockchain via Web3.js to trigger smart contract functions
 - User notifications for milestone achievements, fund release, or refunds

2. Backend (Middleware/API Layer)

- **Technology Stack:** Node.js, Express.js, MongoDB (optional for metadata), IPFS (for decentralized storage)
- **Functionality:**
 - Manages off-chain data such as user profiles, campaign descriptions, images/videos
 - Communicates with smart contracts using Web3.js and Ethereum JSON-RPC API
 - Stores IPFS hashes for campaign media and maps them with campaign IDs
 - Performs validation of form inputs before interacting with smart contracts

- Provides REST APIs for accessing campaign data not stored on the blockchain

3. Blockchain Layer (Smart Contract Layer)

- **Technology Stack:** Solidity, Ethereum (or any EVM-compatible blockchain like Polygon, BNB Chain), Hardhat/Truffle
- **Smart Contracts:**
 - **CampaignFactory Contract:** Allows creation of new campaigns with metadata, funding goal, and deadline.
 - **Campaign Contract:**
 - Stores contributions securely with contributor addresses and amount
 - Handles milestone-based fund release with automated checks
 - Refunds contributors if the campaign fails to meet its goal
 - Provides immutable records of all transactions and campaign states
- **Security Features:**
 - Reentrancy guards to prevent repeated withdrawal attacks
 - Input validation and access control via modifiers (e.g., onlyOwner)
 - Event logging for transparency and auditability of every transaction

4. Storage Mechanism

- **On-Chain:** Transactional data such as contribution amounts, contributor addresses, fund disbursements
- **Off-Chain:** Media files (images/videos of campaigns) stored on IPFS or cloud storage. Metadata (e.g., campaign descriptions) stored off-chain to optimize gas fees

Workflow Overview

1. **Campaign Creation:** A user connects their wallet and fills in campaign details. Smart contracts are deployed with campaign-specific parameters.
2. **Contribution:** Donors select a campaign and send funds directly to the campaign contract.
3. **Milestone Verification:** Once a milestone is reached, the owner requests a fund release. A voting or verification process (optional) may be used.
4. **Fund Disbursement:** If the milestone is valid, smart contracts release funds to the campaign creator automatically.
5. **Refund Mechanism:** If the goal is not met by the deadline, contributors can withdraw their funds securely from the smart contract.

4.2 System Architecture

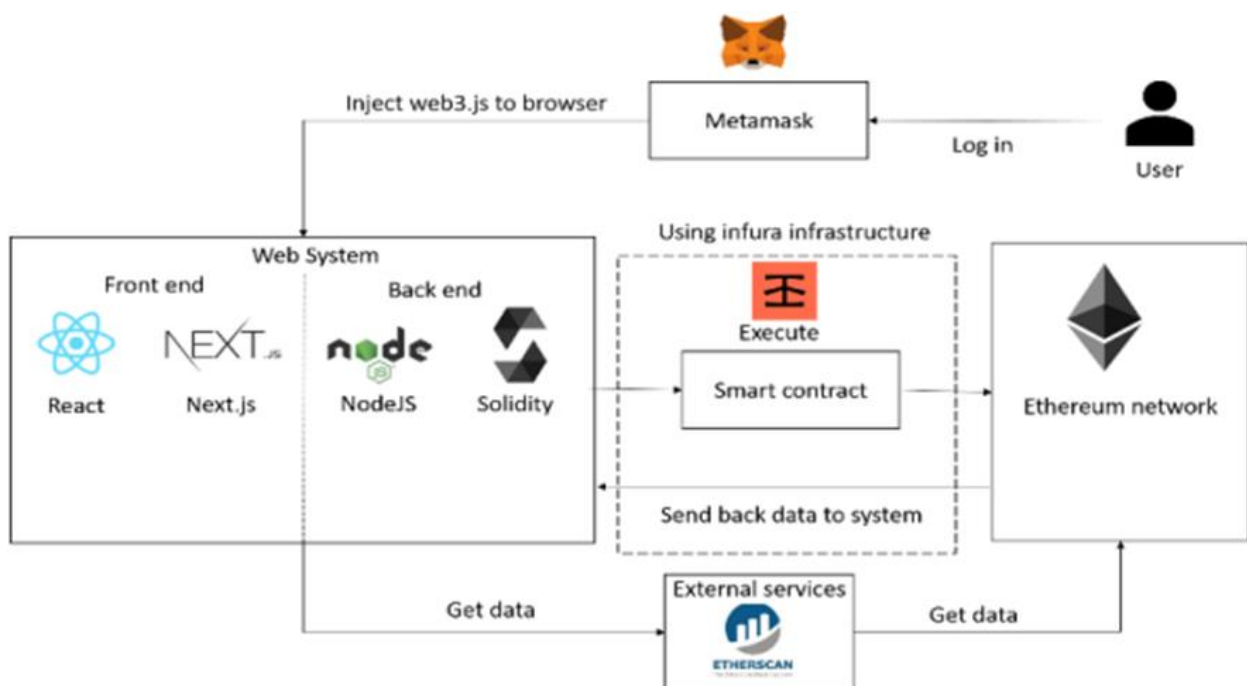


Figure: 4.2.1 System Architecture Design

Figure 4.2.1 illustrates the architecture of a blockchain-based crowdfunding platform integrating Web3 technologies for decentralized interactions. The system begins with the user logging in via the MetaMask wallet, which injects web3.js into the browser to enable communication with the Ethereum blockchain. The web system consists of

a frontend built with React and Next.js, and a backend developed using Node.js and Solidity for smart contract logic. When a user interacts with the platform (e.g., creating or contributing to a campaign), the smart contracts are executed through the Infura.

The smart contracts perform the necessary operations, such as validating transactions or disbursing funds, and send the response back to the system. Additionally, Etherscan is used as an external service to fetch blockchain data for transparency and verification. This architecture ensures secure, trustless, and transparent crowdfunding operations by leveraging decentralized smart contract execution and user-friendly interfaces.

4.3 Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation that illustrates how data moves through a system. It visually describes how input data is processed, stored, and transformed into output.

DFDs help in:

- Understanding system workflow.
- Identifying how different components interact.
- Detecting inefficiencies and optimizing system design.

Purpose of Data Flow Diagram in a Crowdfunding Platform Using Blockchain:

- **Visualize Decentralized Workflow:** Illustrates how users interact with the system via wallets (e.g., MetaMask) and how data flows across the blockchain and web components.
- **Track Transaction Flow:** Shows how user inputs (e.g., donations, campaign creation) trigger smart contract execution and how responses return through decentralized infrastructure.
- **Enhance System Transparency and Security:** Helps identify trust points, secure transaction checkpoints, and ensures clarity in fund flow and smart contract logic.

- **Optimize Architecture:** Aids developers in detecting potential inefficiencies or integration issues across frontend, backend, and blockchain layers.

4.3.1 Data Flow Diagram (Level 0)

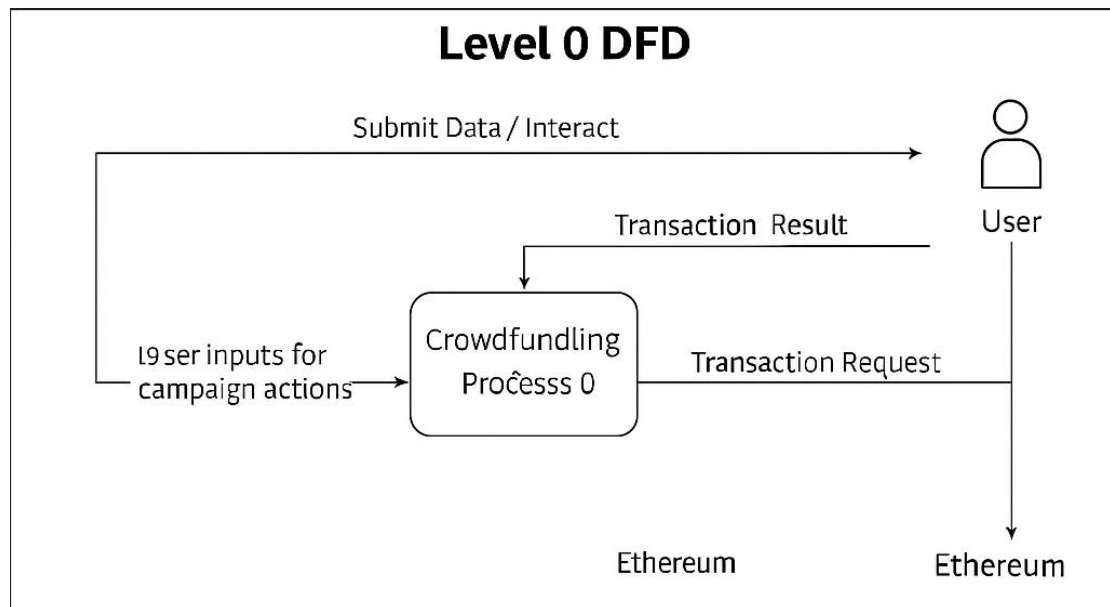


Fig 4.3.1.1 Data Flow Diagram (Level 0)

The **Level 0 Data Flow Diagram** (DFD) for the blockchain-based crowdfunding platform provides a high-level overview of the entire system, illustrating how users interact with the platform through a browser plugin like MetaMask to authenticate and initiate transactions. At this level, the system is treated as a single process that handles inputs (like user logins, campaign creation, and fund transfers) and outputs (such as transaction confirmations and campaign updates), interfacing with the Ethereum network via smart contracts and external services like Etherscan for verification. It simplifies the complex architecture into one core process, highlighting the primary data exchanges between external entities (users, blockchain, and services) and the system itself. The diagram emphasizes how user actions—such as starting a campaign, contributing funds, or withdrawing collected amounts—trigger backend operations that are securely recorded on the blockchain. By leveraging smart contracts, the platform ensures that campaign rules and fund disbursement are automatically enforced without the need for intermediaries. Additionally, transparency is enhanced by integrating Etherscan, which enables users to verify the authenticity and status of blockchain

transactions in real-time. This level provides a foundational understanding of the system's objective: to offer a secure, decentralized, and transparent solution for global crowdfunding.

4.3.2 Data Flow Diagram (Level 1)

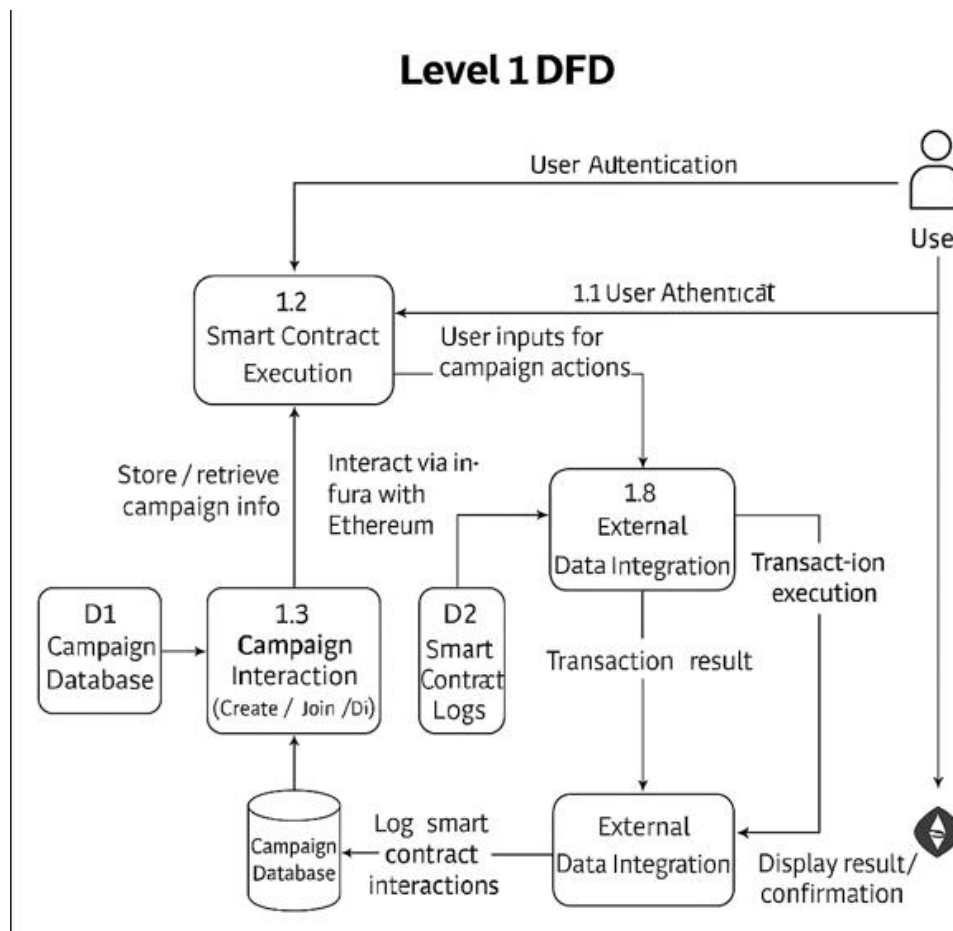


Figure: 4.3.2.1 : Data Flow Diagram (Level 1)

The Level 1 DFD breaks down the internal processes of the blockchain-based crowdfunding platform into more detailed and specific components, highlighting the underlying technological architecture and data interactions within the system. The front-end, developed using React and Next.js, offers an intuitive user interface where users can create campaigns, contribute funds, and monitor campaign progress. Upon accessing the platform, users log in through MetaMask, which injects the Web3.js library into their browsers, enabling secure communication with the Ethereum blockchain. The user requests are then processed by the back-end logic, which is powered by Node.js and integrated with Solidity smart contracts deployed on the blockchain.

4.4 Activity Diagram

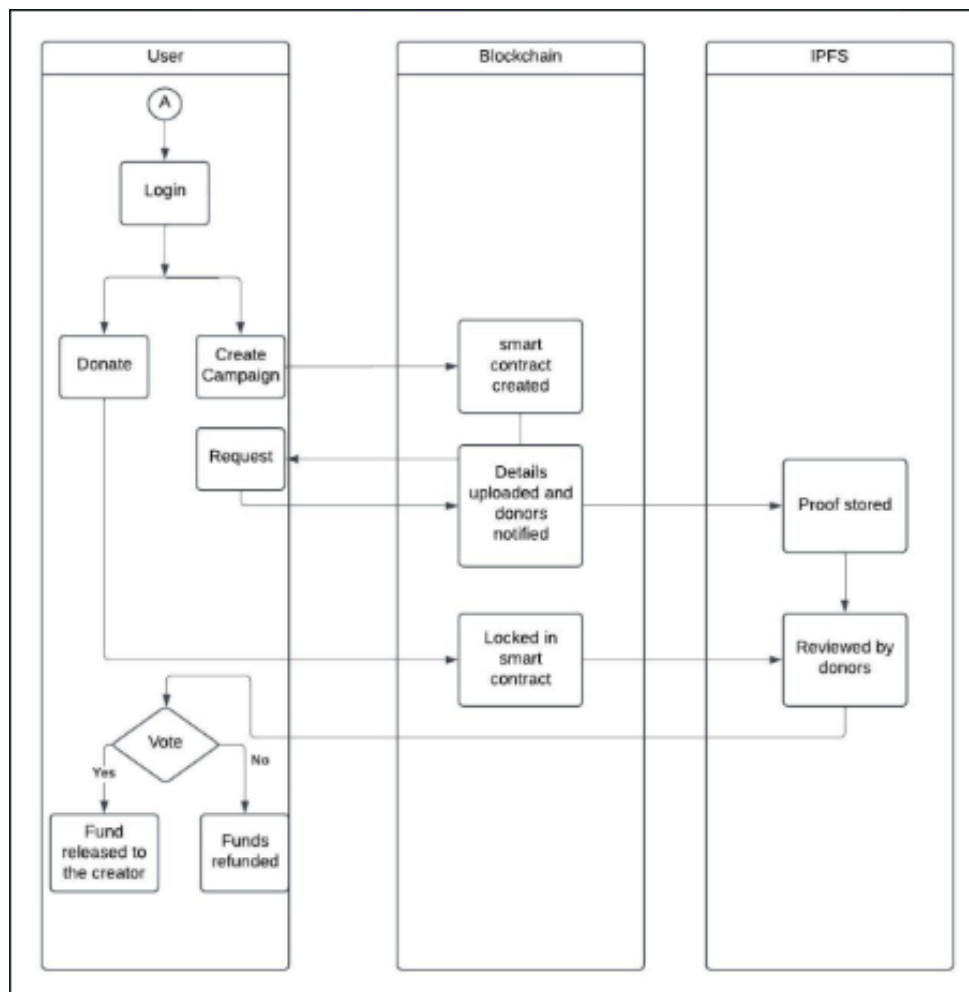


Figure 4.4.1: Activity Diagram

The image illustrates a detailed Data Flow Diagram (DFD) of a blockchain-based crowdfunding platform, capturing the flow of data between various system components and user interactions in a decentralized environment. The process begins when users access the platform and log in via MetaMask, a browser extension that acts as a cryptocurrency wallet and enables blockchain interaction by injecting Web3.js into the browser. Once authenticated, users can initiate various actions such as launching fundraising campaigns, contributing to existing ones, or viewing campaign details. These actions are handled through the platform's web system, which consists of a React and Next.js front end responsible for rendering the user interface, and a Node.js and Solidity-based back end that manages application logic and smart contract functionality.

When a user performs a transaction, a smart contract is triggered and executed through the Infura infrastructure, which acts as a gateway to the Ethereum blockchain. This ensures that transactions are immutable, secure, and trustless. Once processed, the blockchain returns confirmation and relevant data back to the system, which is displayed to users in real-time. Additionally, external services like Etherscan are integrated to retrieve and verify blockchain transaction details, providing transparency and allowing users to independently audit campaign activities. This decentralized architecture ensures high data integrity, removes the need for intermediaries, reduces operational costs, and builds user trust by offering a transparent and tamper-proof crowdfunding ecosystem that can operate on a global scale.

4.5 Use Case Diagram

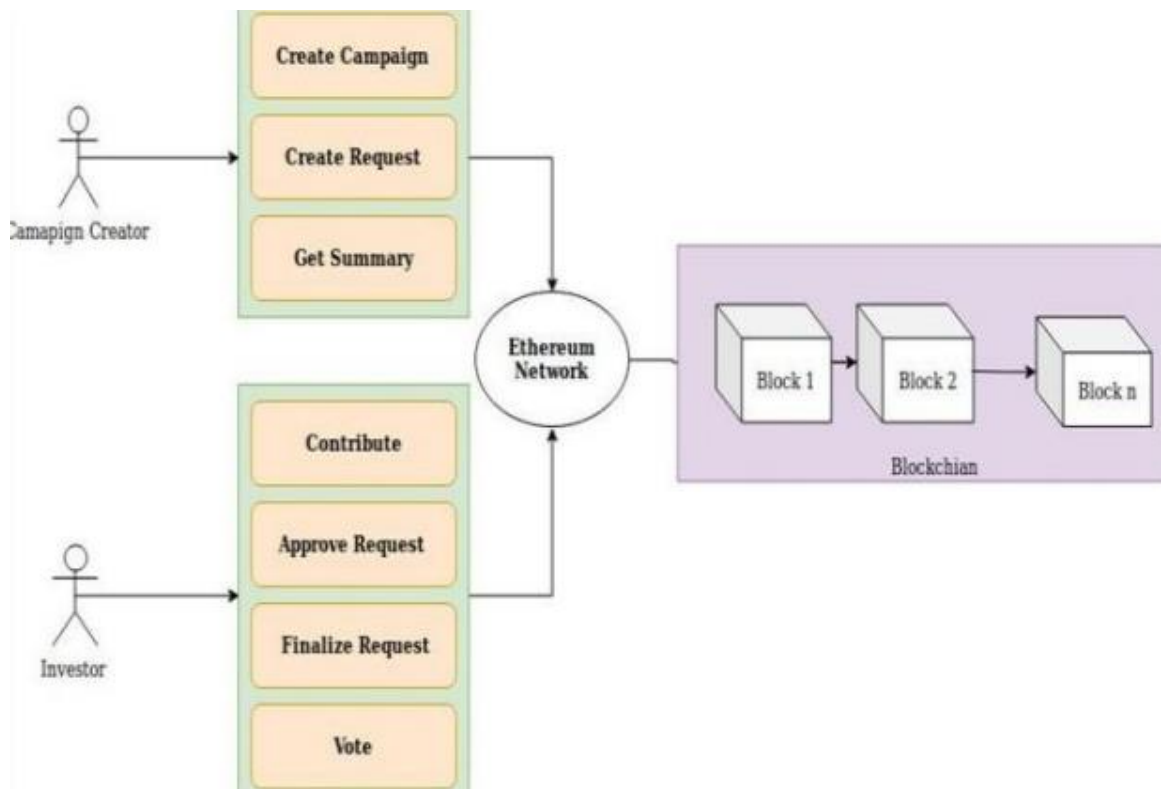


Figure 4.5.1: Use Case Diagram

Figure 4.5.1 provides a comprehensive overview of the operational workflow of a decentralized crowdfunding platform built on the Ethereum blockchain. It clearly illustrates the roles and actions of two key participants: the **Campaign Creator**, who initiates and manages fundraising campaigns, and the **Investor**, who contributes funds and participates in governance. The Campaign Creator begins by performing tasks such as *Create Campaign*, which registers a new fundraising campaign on the Ethereum network using smart contracts. They can also *Create Request* to utilize raised funds for specific needs and *Get Summary* to retrieve real-time campaign data for transparency and reporting. On the other hand, the Investor engages by *Contributing* funds to a campaign, which is securely logged into the blockchain. Once a spending request is made, Investors can *Approve Request* if they agree with how the funds are intended to be used, *Finalize Request* once consensus is reached, and *Vote* on any governance issues. These actions are mediated through the Ethereum network, ensuring that every transaction or request is validated, recorded, and stored as a new block in the blockchain (*Block 1, Block 2,..., Block n*), creating an immutable and auditable record. This architecture enforces transparency, decentralization, and security while enabling smart contract automation to eliminate intermediaries and reduce operational overhead. As a result, the platform fosters greater trust, community participation, and accountability within the crowdfunding ecosystem.

Actors in the System:

1. **Campaign Creator:** The primary actor responsible for creating and managing fundraising campaigns on the platform.
2. **Investor:** The secondary actor who contributes to campaigns, participates in fund utilization decisions, and monitors campaign progress.

Use-Cases and Their Descriptions:

- **Create Campaign:** The Campaign Creator initiates a new crowdfunding campaign by defining details such as title, description, goal amount, and duration.
- **Create Request:** The Campaign Creator submits requests to withdraw or allocate raised funds for specific purposes.
- **Get Summary:** Allows users to view a summary of the campaign, including total contributions, number of investors, and status updates.
- **Contribute:** The Investor sends funds to a chosen campaign using MetaMask and Web3, recorded on the Ethereum blockchain.
- **Approve Request:** Investors review and approve fund utilization requests submitted by the Campaign Creator.
- **Finalize Request:** Once a majority of investors approve a request, the Campaign Creator can finalize it, triggering fund release.
- **Vote:** Investors can vote on critical campaign decisions, enabling a decentralized governance process.
- **Smart Contract Execution:** The system leverages smart contracts to securely and transparently execute all actions related to campaigns and contributions.
- **Blockchain Recording:** Every transaction and interaction is recorded on the Ethereum blockchain for transparency and immutability.

This diagram and use-case model ensure clarity in system behavior, establishing a decentralized, secure, and transparent environment that enhances trust and accountability between campaign creators and investors.

4.6 Sequence Diagram

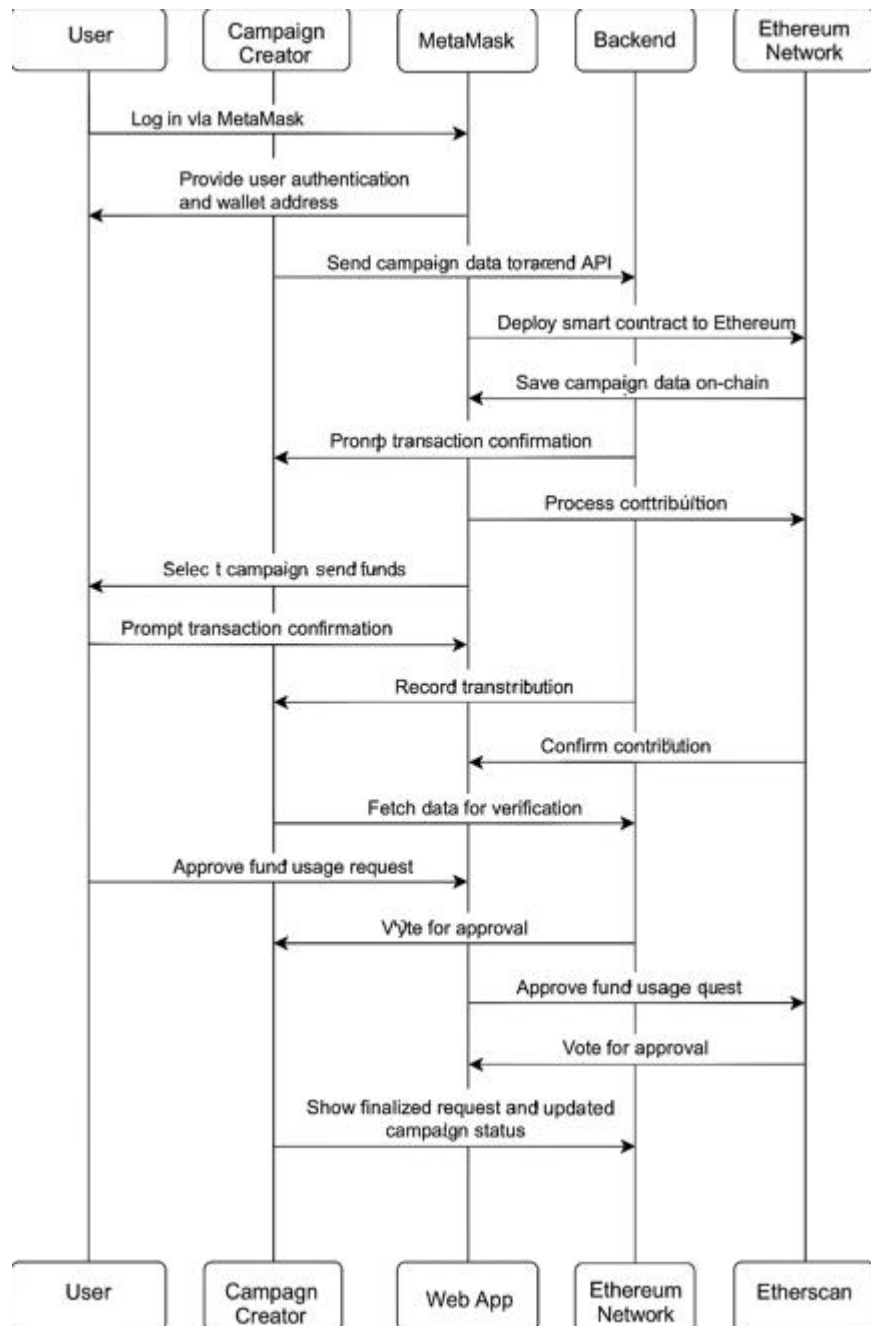


Figure 4.6.1: Sequence Diagram

Figure 4.6.1 presents a comprehensive **sequence diagram** for a blockchain-based crowdfunding platform, illustrating the dynamic interaction between various components involved in the system, including the **User (Investor or**

Campaign Creator), **Web Interface**, **Web3.js**, **Smart Contract**, and the **Ethereum Network**. The process begins when the user logs into the application through the web interface and connects their MetaMask wallet, allowing Web3.js to inject Ethereum-related capabilities into the browser. Once authenticated, the user can perform actions such as creating a campaign, contributing funds, or approving requests depending on their role. The Web Interface communicates with the Smart Contract deployed on the Ethereum blockchain, using Web3.js to pass the transaction data. The Smart Contract, written in Solidity, is executed on the Ethereum Network via Infura infrastructure, which facilitates seamless interaction without requiring users to run a full Ethereum node. Each transaction initiated by the user—whether it's campaign creation, fund transfer, or request approval—is sent to the Ethereum Network, mined, and stored as part of a block on the blockchain, ensuring immutability and transparency. The blockchain then responds with confirmation and relevant data (such as transaction hashes or event logs), which is relayed back to the Web Interface through Web3.js, updating the user interface accordingly. This sequence ensures that all activities on the crowdfunding platform are secure, decentralized, and transparently recorded on the blockchain.

4.7 Summary

The system design phase defines the architecture and workflow of the Handwritten Text Recognition System, ensuring efficiency, accuracy, and scalability. The system consists of multiple modules, including the user interface, preprocessing, feature extraction, text recognition, post-processing, and output generation. It follows a structured pipeline where input images undergo preprocessing, feature extraction using CNNs, recognition using RNNs with CTC loss, and refinement through spell-checking before displaying the final text. Key design considerations include assumptions about input quality, dependencies on deep learning and image processing libraries, and constraints such as computational resources and handwriting variability. Various diagrams, such as Data Flow Diagrams, Activity Diagrams, and Use Case Diagrams, visually represent the system's operations.

CHAPTER 5

IMPLEMENTATION

CHAPTER 5

IMPLEMENTATION

5.1 Introduction

The Crowdfunding Platform is a decentralized web application (dApp) that empowers users to initiate, support, and manage fundraising campaigns through the Ethereum blockchain. By eliminating intermediaries, the system provides transparent and tamper-proof fundraising. The platform is designed with an intuitive user interface, seamless blockchain interaction, and secure fund handling using smart contracts. It also ensures the decentralized storage of campaign-related data, maintaining integrity and availability. This project utilizes Solidity for writing the logic of smart contracts and JavaScript for creating responsive and dynamic user interfaces. It integrates cutting-edge tools and libraries like React for component-based UI development, TailwindCSS for styling, Ethers.js for blockchain communication, and Hardhat for smart contract development. Campaign data such as images and descriptions are stored in IPFS via Pinata, promoting decentralized and persistent storage. The Sepolia testnet is used for testing and simulating real blockchain interactions.

5.2 Programming Language Selection

The development of the crowdfunding platform involves the use of the following programming languages:

- **Solidity:** Used for writing smart contracts deployed on the Ethereum blockchain. Solidity enables secure and reliable execution of functions related to campaign creation, donations, and fund withdrawals. It is a statically typed, contract-oriented language optimized for the Ethereum Virtual Machine (EVM).

JavaScript: Used extensively for frontend development and for integrating blockchain functions using Ethers.js.

- **Markdown:** Utilized in documentation, Jupyter Notebooks, and project reports for formatting content, explanations, and technical analysis. Markdown enhances the readability and structure of technical documents.

Solidity's seamless integration with Ethereum, JavaScript's universal web presence, and Markdown's documentation capabilities make them ideal choices for implementing and presenting this project.

5.3 Platform/Framework

The system leverages a modern stack of development platforms and frameworks to ensure scalability, modularity, and robustness:

- **React:** A powerful JavaScript library for building user interfaces. React provides a component-based architecture, facilitating code reuse and faster development of dynamic, interactive UIs.
- **TailwindCSS:** A utility-first CSS framework that allows for rapid design of clean and responsive user interfaces without writing custom CSS.
- **Hardhat:** A development environment for compiling, deploying, testing, and debugging Ethereum smart contracts. Hardhat allows developers to simulate blockchain behavior locally and provides comprehensive debugging tools.
- **Ethers.js:** A lightweight JavaScript library for interacting with Ethereum blockchain and smart contracts. It simplifies blockchain communication through APIs and enhances wallet integration.
- **Pinata (IPFS):** A decentralized file storage service built on IPFS. Pinata enables uploading, pinning, and accessing campaign images and metadata in a decentralized, persistent manner.
- **Sepolia Testnet:** A secure Ethereum test network used to test contracts without using real ETH. Sepolia replicates mainnet behavior and is supported by tools like MetaMask and Etherscan.

These platforms and frameworks collectively facilitate seamless development, deployment, testing, and interaction with the decentralized crowdfunding application.

5.4 Tools

The following tools are used for project development: Multiple development and deployment tools were used to ensure smooth implementation and testing:

Visual Studio Code (VS Code): The primary code editor used for writing and managing frontend and smart contract files. VS Code offers excellent extension support for Solidity, JavaScript, and TailwindCSS.

- **Hardhat:** A robust framework for compiling, deploying, and testing Solidity contracts. It supports local blockchain simulation and contract verification scripts.
- **Pinata Gateway:** A service for interfacing with IPFS, allowing campaign metadata and images to be stored and accessed in a decentralized way.
- **MetaMask:** A browser extension wallet used to manage user accounts, authorize transactions, and interact with deployed contracts.
- **Etherscan:** A blockchain explorer that verifies contract deployment, shows transactions, and aids in debugging by tracking smart contract activity on the Sepolia testnet.

These tools contribute to efficient development cycles, reliable deployment, and transparent verification of smart contract operations.

5.5 Implementation

The implementation is modular and organized into three main parts: frontend interface, smart contract backend, and decentralized storage integration.

5.5.1 Frontend

Key Functionalities:

- MetaMask integration for wallet connection.
- Campaign creation form where users upload images and enter metadata.
- Dynamic display of all campaigns with visual donation progress.
- Real-time ETH donation form linked to smart contract.

- Conditional withdrawal option for campaign creators.

Technologies Used:

- **React:** Enables UI modularization using reusable components like CampaignCard, CampaignForm, and Navbar.
- **TailwindCSS:** Ensures consistent styling and responsiveness across devices.
- **Ethers.js:** Facilitates wallet connection, contract function calls, and event listening.

This combination allows the platform to provide real-time updates, a user-friendly experience, and secure transaction processing.

```
return (  
  <header className="flex justify-between items-center bg-gradient-to-r from-indigo-600 to-purple-600 text-white px-8 py-4 fixed top-0" >  
    { /* Logo */ }  
    <Link to="/" className="text-3xl font-bold hover:opacity-90 transition">  
      FundIt  
    </Link>  
  
    { /* Navigation */ }  
    <nav className="flex items-center gap-8">  
      <Link  
        to="/my-campaigns"  
        className="text-white font-medium hover:bg-white hover:text-indigo-600 px-4 py-2 rounded-lg transition"  
      >  
        My Campaigns  
      </Link>  
  
      { /* Create Campaign Dialog */ }  
      <Dialog>  
        <DialogTrigger asChild>  
          <Button variant="outline" className="bg-white text-indigo-600 hover:bg-gray-100">  
            Create Campaign  
          </Button>  
        </DialogTrigger>  
        <DialogContent className="bg-white rounded-lg p-6 w-full max-w-lg shadow-xl">  
          <DialogHeader>  
            <DialogTitle className="text-indigo-600">Create Campaign</DialogTitle>  
          </DialogHeader>  
        </DialogContent>  
      </Dialog>  
    </nav>  
  </header>  
)
```

Fig 5.5.1.1 Implementation Of Frontend Pseudocode

The Header component serves as the primary navigation and interaction point for users on your blockchain-based crowdfunding platform. It integrates wallet connection using a custom useWallet hook, allowing users to connect their Ethereum wallet (on the Sepolia network), view their balance (converted from Wei to ETH), and access campaign-related features. The component also provides navigation links like "My Campaigns" and displays the user's current wallet balance once connected. Additionally, it features a dialog-based UI for

creating new crowdfunding campaigns, utilizing Tailwind CSS and shadcn/ui components for a clean, modern design.

The campaign creation form allows users to input various campaign details including title, description, funding goal, duration, genres, and a banner image, which is uploaded to IPFS via Pinata. Upon submission, the form interacts with a smart contract (retrieved using a custom useContract hook) to create a new campaign on the blockchain through the createCampaign function in the campaignFactory contract. The component also manages genre selection, image uploading, and real-time balance updates through React state and useEffect.

5.5.2 Backend

The core backend logic is implemented using Solidity smart contracts. Key operations like campaign creation, donation management, and fund withdrawal are managed within the contract.

Smart Contract Features:

- Campaign struct storing metadata including title, description, target amount, deadline, and IPFS hash.
- createCampaign() function to initialize and store campaigns.
- donateToCampaign() function to accept and record donations.
- withdrawFunds() function restricted to campaign owner once the goal is met.
- Events emitted on campaign creation and donation to facilitate frontend updates.

Security Measures:

- Modifier checks for owner-only withdrawals.
- Input validations and time checks.
- Use of require() to enforce contract conditions.

The contract is deployed to Sepolia using Hardhat scripts and validated on Etherscan for transparency.

```
function getContributors() public view returns (address[] memory) {
    return contributors;
}

// working in remix
function getWithdrawRequestInfo(uint _index) public view returns(address[] memory , uint[] memory,uint256,uint256,uint256,string memory)
{
    uint[] memory res = new uint[](contributors.length);
    for(uint i=0;i<contributors.length;i++)
        res[i] = (uint(requests[_index].votingResult[contributors[i]]));
    return(contributors,res, requests[_index].amt , requests[_index].date , uint(requests[_index].status), requests[_index].ipfsproofadd);
}

function getWithdrawRequestVotingResult(uint256 _index, address _contributor) external view returns (uint8) {
    require(_index < requests.length, "Invalid request index");
    return uint8(requests[_index].votingResult[_contributor]);
}

modifier isCampaignOpen() {
    require(status == CampaignStatus.Opened, "This campaign can't accept donations");
    _;
}

modifier isValidTransaction() {
    require(msg.value <= goal - fundRaised, string.concat("Donations should be less than or equal to ", Strings.toString(goal - fundRaised)));
    require(msg.value != 0, "Invalid donation");
    _;
}
```

```
function getCampaignCard() external view returns(string memory , string memory , uint , string memory , uint , uint,uint,string memory)
{
    return (name,description,goal,pic,uint(status),durationindays,fundRaised,genres);
}

function getCampaignDetails1() external view returns(string memory , string memory , string memory , uint , uint ,uint , uint , uint ,address[] memory , address )
{
    return (
        name,
        description,
        pic,
        goal,
        uint(status),
        durationindays,
        fundRaised,
        startdate,
        contributors,
        owner
    );
}
```

```
function getCampaignDetails2() external view returns(uint[] memory, uint[] memory , uint[] memory , string[]memory)
{
    uint[] memory _status = new uint[](requests.Length);
    uint[] memory amt = new uint[](requests.Length);
    string[] memory ipfsproofadd = new string[](requests.Length);
    uint[] memory _contributions = new uint[](contributors.Length);

    // Fill _contributions array with the contributions
    for(uint i = 0; i < contributors.Length; i++) {
        _contributions[i] = contributions[contributors[i]];
    }

    // Fill the withdraw request details
    for(uint i = 0; i < requests.Length; i++) {
        _status[i] = uint(requests[i].status);
        amt[i] = requests[i].amt;
        ipfsproofadd[i] = requests[i].ipfsproofadd;
    }
    return(_contributions,_status,amt,ipfsproofadd);
}
```

Figure 5.5.2.1 Backend Pseudocode

This Solidity smart contract implements a blockchain-based crowdfunding platform, allowing users to contribute ETH to campaigns, and campaign owners to request fund withdrawals based on contributor voting. The contract starts with necessary imports and defines two enums—VotingStatus and CampaignStatus—for managing withdrawal voting and campaign lifecycle respectively. The core data structure is withdrawRequest, which includes a date, amount, IPFS proof, final voting status, and a mapping of individual contributor votes. The contract keeps track of campaign details like goal, description, image, amount raised, start and end dates, contributors, and withdrawal requests.

Upon deployment, the campaign initializes its parameters and enters an “Opened” status, accepting contributions. Contributors can donate ETH, and their addresses and amounts are tracked. Once the fundraising goal is met, the campaign status updates to “Completed.” The campaign owner can then create a withdrawal request, including an IPFS link for proof, which each contributor must vote on. Based on voting results, the request is either accepted (and funds transferred to the owner) or rejected. The contract includes functions to fetch campaign and request details, vote on withdrawals, and refund contributors proportionally if necessary. Additional view functions provide front-end support by exposing campaign metadata, contributor lists, and request histories.

5.6 Summary

This smart contract implements the backend logic for a decentralized crowdfunding platform on the Ethereum blockchain. It allows a campaign owner to create a campaign with a funding goal, duration, and description, while users can contribute funds. All contributions are tracked, and once the goal is reached or the campaign ends, the owner can create withdrawal requests. Contributors then vote to approve or reject each withdrawal request, ensuring transparency. Funds are only transferred if the majority approves. The contract also supports refunds if needed and provides multiple view functions to fetch campaign details, contributions, and withdrawal history, making it fully interactable with a frontend via Web3.

CHAPTER 6

RESULTS AND DISCUSSION

CHAPTER 6

RESULTS AND DISCUSSION

This chapter presents the results obtained from the blockchain-based crowdfunding platform. The system was tested using multiple fundraising campaigns and donation scenarios on the Ethereum test network. Performance evaluation was carried out based on user interactions, transaction confirmation, and smart contract execution. The discussion includes an analysis of system functionality, security, and transparency, followed by insights into its strengths and areas for future improvement.

6.1 Results

The blockchain-based crowdfunding platform successfully facilitates the creation, funding, and tracking of campaigns through a decentralized architecture. The platform leverages MetaMask for user authentication, React and Next.js for user interaction, and Solidity smart contracts for automation and fund management.

The results indicate that:

- The system allows users to create campaigns and receive funds transparently through secure smart contract logic.
- Donors can seamlessly contribute to campaigns, and transaction hashes are verifiable via Etherscan, ensuring traceability.
- Smart contracts automatically enforce rules, such as disbursing funds only after reaching the set target, reducing the risk of fraud.
- The use of blockchain technology enhances user trust by offering immutable records and real-time updates.

Minor delays in transaction confirmation were observed during high network activity, highlighting the need for potential scalability improvements using Layer 2 solutions.



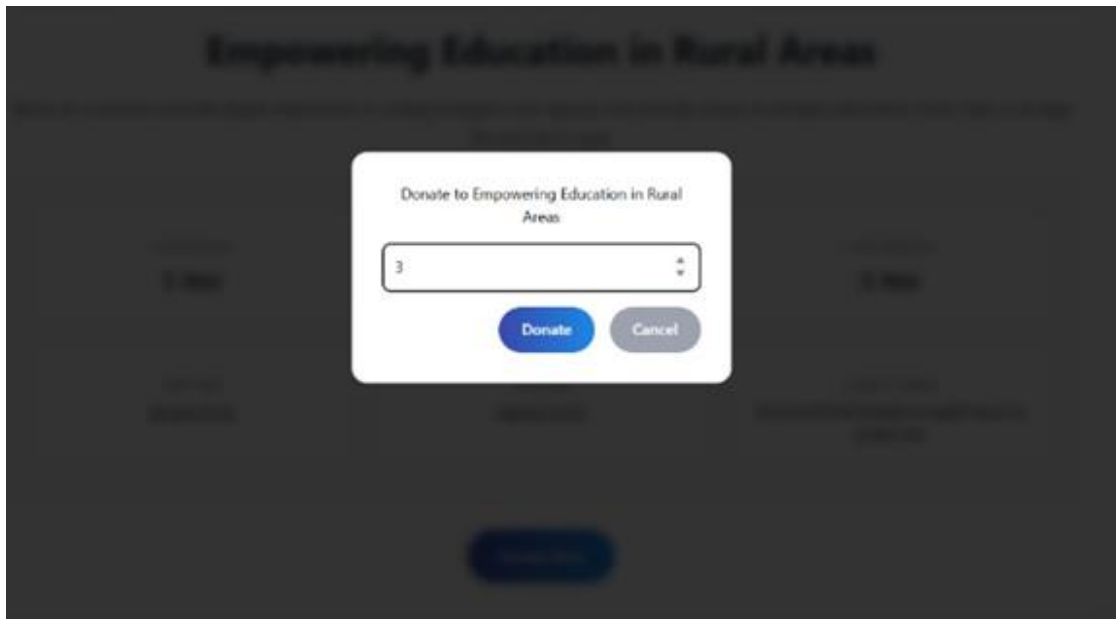


Figure 6.1.4 Donating to Campaign

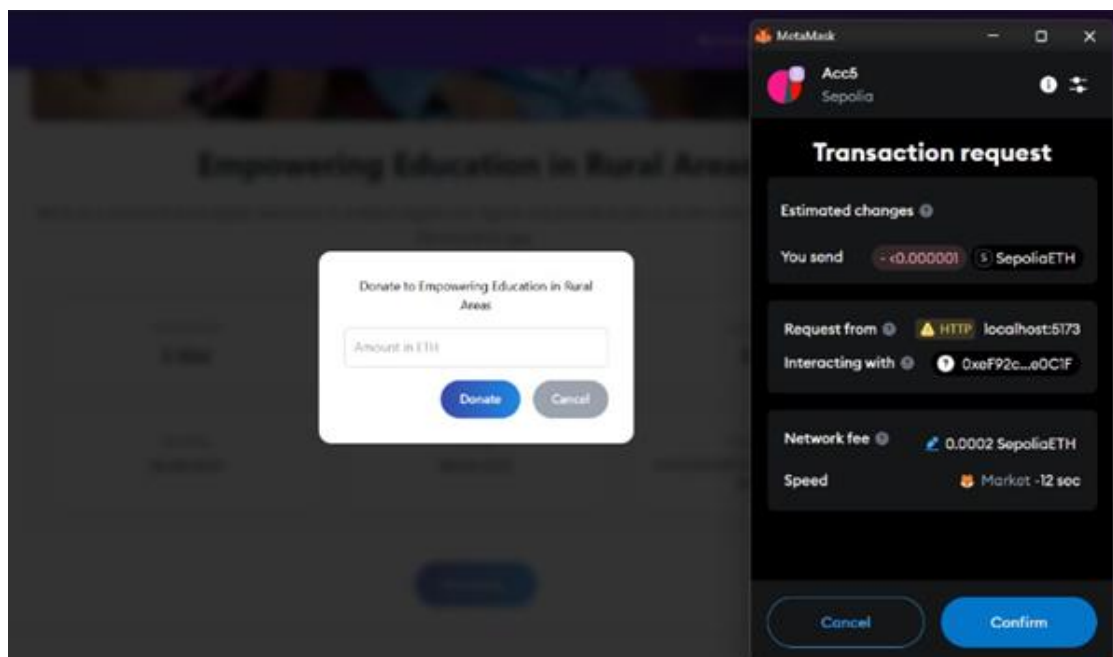


Figure 6.1.5 Transaction Request



Figure 6.1.6 Donating



Figure 6.1.7 Create request by campaign manager

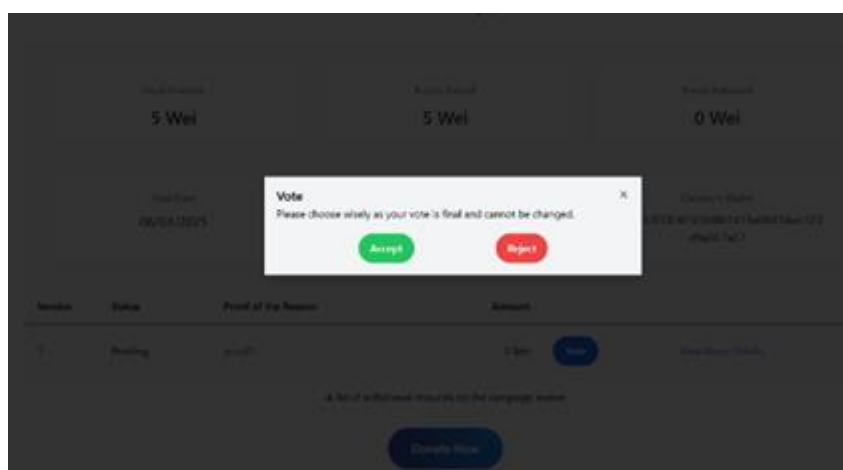


Figure 6.1.8 Voting to the Campaign

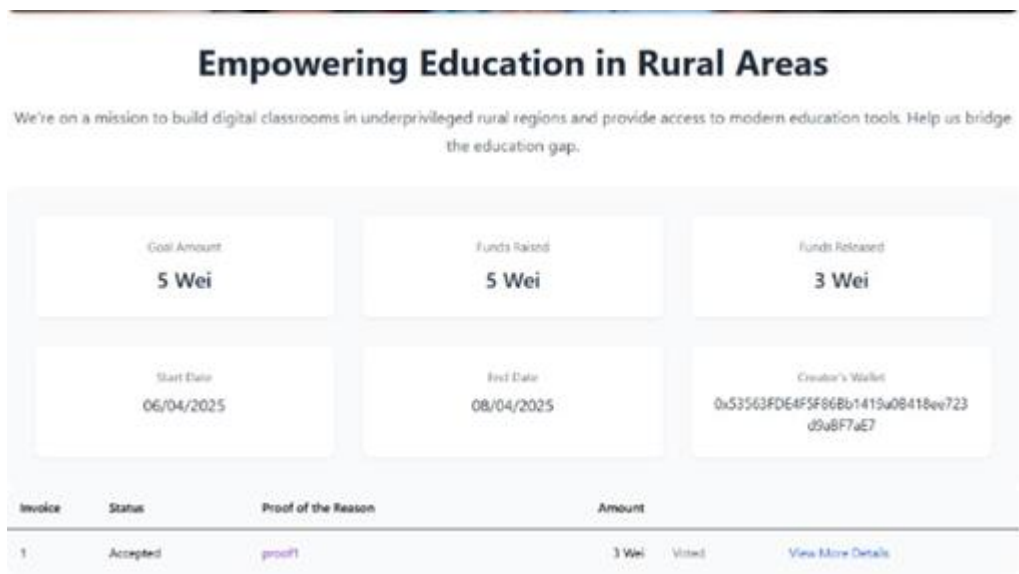


Figure 6.1.9 Funds Released

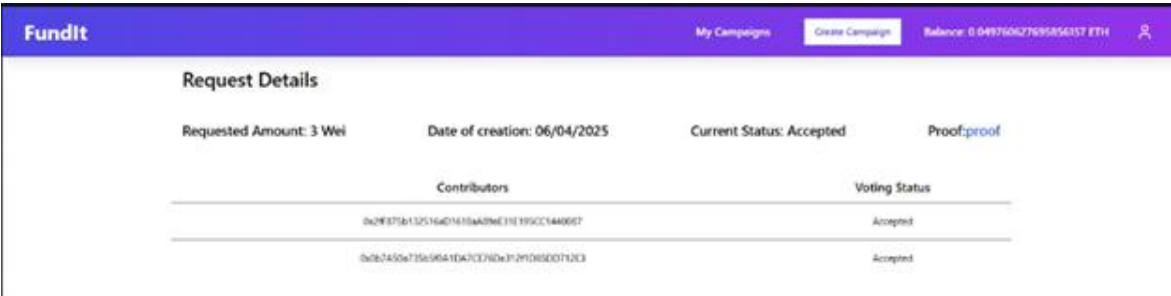


Figure 6.1.10 Details of the request

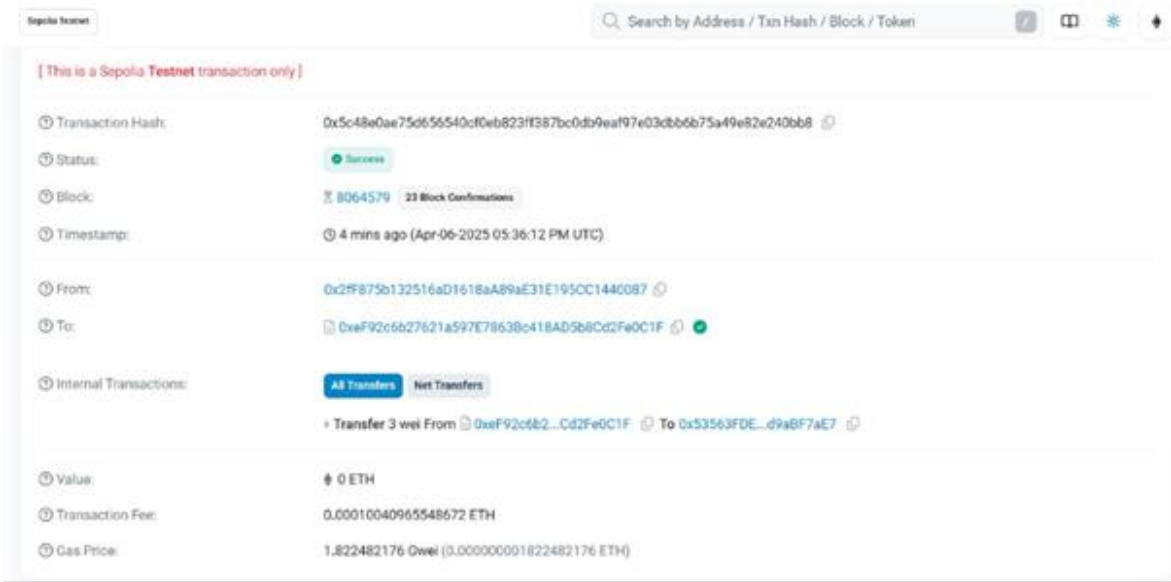


Figure 6.1.11 Recorded in EtherScan

6.2 Summary

The blockchain-based crowdfunding platform effectively demonstrates the use of decentralized technologies to support transparent, secure, and automated fundraising. Through the integration of MetaMask, smart contracts, and Ethereum blockchain infrastructure, the system enables users to create and support campaigns without intermediaries. Results show successful handling of campaign creation, donations, and fund disbursements with verifiable transactions. The platform ensures transparency through Etherscan integration and enforces predefined rules via smart contracts. While the system performed well under typical loads, minor latency was noted during high network activity, indicating room for optimization. Overall, the platform provides a reliable, trustless environment for digital crowdfunding with potential for scalability and real-world adoption.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

This project successfully developed a decentralized crowdfunding platform using blockchain technology, offering a secure, transparent, and trustless environment for raising and managing funds. By leveraging smart contracts, the system ensures automated fund distribution, eliminates the need for intermediaries, and maintains an immutable record of transactions. The platform enhances accountability and donor confidence through publicly verifiable campaign data while providing a user-friendly interface for both campaign creators and contributors. Though effective in its current form, the platform's long-term success depends on addressing challenges related to scalability, regulatory compliance, and broader user adoption.

7.2 Future Scope

1. Integration of Decentralized Identity (DID) and KYC

- Implementing decentralized identity frameworks to authenticate users without compromising privacy.
- Ensuring compliance with KYC/AML regulations using blockchain-based identity verification mechanisms.

2. Enhanced Fraud Detection and Trust Mechanisms

- Integrating AI-based fraud detection to flag suspicious activities or fake campaigns.
- Introducing a **reputation system** or token-based voting to evaluate and rank campaigns based on trustworthiness

3. Cross-Chain Support and Multi-Currency Payments

- Enabling interoperability with multiple blockchains (e.g., Ethereum, Solana, Polygon) to broaden user base.

- Supporting various cryptocurrencies and stablecoins to offer flexibility in contributions.

4. Governance and DAO Implementation

- Establishing **Decentralized Autonomous Organization (DAO)** for community-led governance.
- Allowing token holders to vote on platform upgrades, dispute resolution, and project approvals.

5. Mobile App Development and Real-Time Notifications

- Creating a mobile version of the platform for seamless access and real-time transaction alerts.
- Incorporating wallet integration and biometric security for enhanced user experience.

6. Legal and Regulatory Framework Expansion

- Collaborating with regulatory bodies to ensure compliance in different jurisdictions.
- Developing a legal framework for campaign legitimacy and dispute handling in blockchain environments

APPENDIX – I

APPENDIX – I

Sustainable Development Goals

The Handwritten Text Recognition (HTR) project aligns with several United Nations Sustainable Development Goals (SDGs) by promoting technological advancements in education, accessibility, and digital transformation. The key SDGs relevant to this project are:

1. **SDG 1: No Poverty** - The platform empowers individuals and communities to raise funds for poverty alleviation initiatives. By offering a decentralized way to access financial support, it helps people in need without relying on traditional banking or intermediaries.
2. **SDG 2: Zero Hunger** - Campaigns targeting food security, nutrition, and agricultural development can leverage the platform to gather donations and resources, thereby supporting efforts to end hunger and promote sustainable food systems.
3. **SDG 3: Good Health and Well-Being** - Healthcare-related fundraising becomes more transparent and efficient through blockchain, enabling people to support medical treatments, mental health services, and emergency aid with verifiable and traceable donations.
4. **SDG 4: Quality Education** - The platform allows educational institutions, students, and NGOs to raise money for tuition, infrastructure, and learning materials. It broadens access to education by reducing the financial gap, especially in underprivileged areas.
5. **SDG 5: Gender Equality** - It supports initiatives that empower women and girls by enabling gender-focused campaigns and ensuring equal opportunity for all genders to start and receive funding through secure and transparent means.
6. **SDG 10: Reduced Inequalities** – By enabling automated handwritten text recognition, the project supports accessibility for people with disabilities, including those with visual impairments and dyslexia. It ensures that digitized text can be used for assistive technologies,

promoting equal opportunities

7. **SDG 16: Peace, Justice, and Strong Institutions** – Efficient OCR technology helps in digitizing legal and historical documents, ensuring transparency, accountability, and easier access to crucial records. This supports the integrity of legal and governance systems.
8. **SDG 17: Partnerships for the Goals** - The platform facilitates collaboration between donors, communities, NGOs, and institutions across the globe, encouraging partnerships that drive collective action towards achieving the SDGs.

This project contributes to a sustainable future by improving accessibility, fostering innovation, and enhancing digital inclusivity.

APPENDIX – II

APPENDIX – II

PAPER PUBLICATION DETAILS

Conference Title:

2nd International Conference on New Frontiers in Communication, Automation, Management and Security (ICCAMS 2025)

Conference Dates: 10th, 11th, and 12th July 2025

Organized by: Presidency University, Bangalore

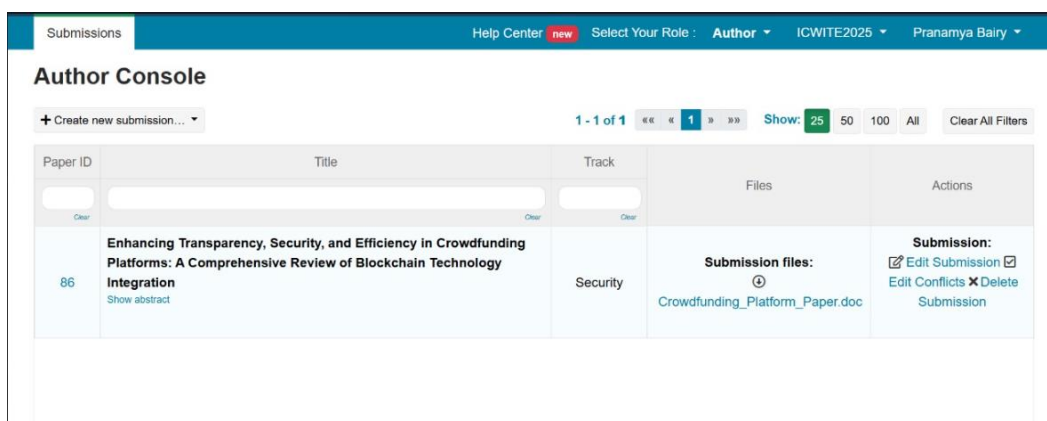
Conference Website: <https://iccams-presidencyuniversity.netlify.app/>

Paper Title: "Crowdfunding Platform using Blockchain"

Submission Status:

The paper titled "Crowdfunding Platform using Blockchain " has been submitted to the ICCAMS 2025 for review and consideration for publication. The paper is currently under review and acceptance is awaited.

Proof of Submission:



The screenshot displays the 'Author Console' interface. At the top, there's a navigation bar with 'Submissions', 'Help Center', 'new', 'Select Your Role : Author', 'ICWITE2025', and 'Pranamy Bairy'. Below this, the 'Author Console' header is followed by a '+ Create new submission...' button and pagination controls showing '1 - 1 of 1'. A 'Show:' dropdown is set to '25', with options for 50, 100, and 'All'. A 'Clear All Filters' button is also present. The main content is a table with columns: Paper ID, Title, Track, Files, and Actions. The first row shows a submission with Paper ID '86', the title 'Enhancing Transparency, Security, and Efficiency in Crowdfunding Platforms: A Comprehensive Review of Blockchain Technology Integration', and the track 'Security'. The 'Files' column lists 'Submission files: Crowdfunding_Platform_Paper.doc'. The 'Actions' column includes links for 'Edit Submission', 'Edit Conflicts', 'Delete Submission', and 'Submission'.

Paper ID	Title	Track	Files	Actions
86	Enhancing Transparency, Security, and Efficiency in Crowdfunding Platforms: A Comprehensive Review of Blockchain Technology Integration	Security	Submission files: Crowdfunding_Platform_Paper.doc	Submission: Edit Submission Edit Conflicts Delete Submission

REFERENCES

REFERENCES

- [1] Freedman et al. "The Foundations of Online Crowdfunding", Equity Crowdfunding for Investors (eds D. M. Freedman and M. R. Nutting), 2015, doi:10.1002/9781118864876.ch1.
- [2] T. Dannberg, "Advantages and Disadvantages with Crowdfunding : - and Who are the Users?", Dissertation, 2017.
- [3] Schwienbacher et al, "Crowdfunding of Small Entrepreneurial Ventures", Handbook Of Entrepreneurial Finance, Oxford University Press. <http://dx.doi.org/10.2139/ssrn.1699183>.
- [4] Macht et al, " The Benefits of Online Crowdfunding for Fund-Seeking Business Ventures", Strategic Change, 2014, 23 (1-2). pp. 1-14. ISSN 10861718.
- [5] Schlueter et al, "Underlying Benefits and Drawbacks of Crowdfunding from the Perspective of Entrepreneurs in Germany", 5th IBA Bachelor Thesis Conference, University of Twente. Available at: http://essay.utwente.nl/67409/1/Schlueter_BA_MB.pdf [Accessed 15 Aug 2018].
- [6] Gabison et al, "Understanding Crowdfunding and its Regulation", 2015, doi:10.2791/562757.
- [7] Ramos et al, S. (2014). "Crowdfunding and the Role of Managers in Ensuring the Sustainability of Crowdfunding Platforms (Rep.)", Publications Office of the European Union. doi:10.2791/76003.
- [8] Cumming et al, "Crowdfunding from Fraudfunding", Max Planck Institute for Innovation & Competition Research Paper No. 16-09. Available at SSRN: <https://ssrn.com/abstract=2828919> or <http://dx.doi.org/10.2139/ssrn.2828919>.
- [9] Mollick et al., "The Dynamics of Crowdfunding: An Exploratory Study (June 26, 2013)", Journal of Business Venturing, Volume 29, Issue 1, January 2014, Pages 1–16.
- [10] Z. Zheng et al, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends", IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, 2017, pp. 557-564.doi: 10.1109/BigDataCongress.2017.85.
- [11] Miraz et al., "Applications of Blockchain Technology beyond Cryptocurrency", Annals

of Emerging Technologies in Computing (AETiC), 2018. 2. 1-6.

[12] Chen et al., "Exploring Blockchain Technology and its Potential Applications for Education. Smart Learning Environments", 5. 10.1186/s40561-017-0050-x.

[13] A. Angrish et al., "A Case Study for Blockchain in Manufacturing: "FabRec": A Prototype for Peer-to-Peer Network of Manufacturing Nodes", *Procedia Manufacturing*, vol. 26, pp. 1180-1192, 2018.

[14] Friedlmaier et al., "Disrupting Industries With Blockchain: The Industry, Venture Capital Funding, and Regional Distribution of Blockchain Ventures", *Proceedings of the 51st Annual Hawaii International Conference on System Sciences (HICSS)*, January 2018.

Available at SSRN:
<https://ssrn.com/abstract=2854756> or <http://dx.doi.org/10.2139/ssrn.2854756>

[15] Alharby et al., "Blockchain Based Smart Contracts: A Systematic Mapping Study", 2017, 125-140. 10.5121/csit.2017.71011.

[16] Delmolino et al., "Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab", 2016, 9604. 79-94. 10.1007/978-3-662-53357-4_6m.

[17] Gebert et al., "Application Of Blockchain Technology In Crowdfunding", 2017, New European.

[18] Ming Li et. al. (2017). CrowdBC: A Blockchain-based Decentralized Framework for Crowdsourcing. *IACR Cryptology ePrint Archive*.

[19] Tsankov, P., Dan, A.M., Cohen, D.D., Gervais, A., Buenzli, F., & Vechev, M.T. (2018). Securify: Practical Security Analysis of Smart Contracts. *CoRR*, abs/1806.01143.

[20] A. Kosba, A. Miller, E. Shi, Z. Wen and C. Papamanthou, "Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts," 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, 2016, pp. 839-85.