A PROJECT REPORT

on

"TokenMaster"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

BY

Rahul Yadav   21053310

UNDER THE GUIDANCE OF
Swagatika Sahoo

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024
May 2020

A PROJECT REPORT
on

“TokenMaster”

Submitted to
KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR’S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

BY

Rahul Yadav 21053310

UNDER THE GUIDANCE OF
Swagatika Sahoo

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAE, ODISHA -751024
May 2022

KIIT Deemed to be University
School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

“TokenMaster“

Submitted by
Rahul Yadav 21053310

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2023-2024, under our guidance.

Date:2024/04/09

Swagatika Sahoo
Project Guide

Acknowledgements

We are profoundly grateful to **Swagatika Sahoo** of **School of Computer Engineering** for her expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

Rahul Yadav

ABSTRACT

TokenMaster is a blockchain-based ticketing platform that aims to transform ticket booking experience for Bollywood concerts. By utilizing blockchain technology, the platform guarantees transparency, security, and fairness in ticket distribution Built with React for the frontend and ether.js for seamless interaction with Ethereum smart contracts, TokenMaster provides users with a user-friendly interface for purchasing tickets using Ether. After a successful transaction, users will receive a unique NFT ticket directly to their MetaMask wallet, acting as proof of ownership and authenticity. To prevent ticket scalping and ensure fair access, TokenMaster enforces a single-ticket purchase limit, deterring bulk buying by bots or scalpers. Through its innovative approach, TokenMaster seeks to redefine the ticketing industry, offering Bollywood enthusiasts a reliable and convenient platform to attend their favorite concerts.

TokenMaster emerges as a pioneering solution in the realm of Bollywood concert ticketing, presenting a blockchain-powered platform tailored to meet the evolving needs of users. The integration of blockchain technology instills trust and integrity into the ticketing process, fostering a transparent and secure environment for transactions. Built on React and ether.js, TokenMaster offers an intuitive frontend interface, streamlining the ticket purchasing experience with seamless Ethereum smart contract interaction. With Ether as the primary mode of payment, users can effortlessly acquire tickets, each minted as a unique NFT and delivered directly to their MetaMask wallet. By enforcing a single-ticket purchase restriction, TokenMaster mitigates the risks associated with ticket scalping, ensuring fair and equitable access to concert tickets. TokenMaster stands poised to redefine the paradigm of Bollywood concert ticketing, setting a new standard of efficiency and reliability for enthusiasts eager to partake in live musical experiences.

Keywords: Blockchain ticketing, NFT tickets, Ethereum smart contracts, Bollywood concerts ticketing, MetaMask wallet integration, NFTainment

Contents

List of Figures

Chapter 1

Introduction

In a world where the digital landscape is evolving rapidly, traditional ticketing systems often struggle to keep pace with the demands of modern consumers. Enter TokenMaster, a groundbreaking project that aims to revolutionize the ticketing industry by harnessing the power of blockchain technology.

TokenMaster provides a novel solution to the challenges faced by both event organizers and ticket buyers, providing a transparent, secure, and efficient platform for booking tickets to Bollywood actors and singers' concerts. TokenMaster utilizes blockchain technology to guarantee integrity, traceability, and decentralization in ticket transactions, paving the way for enhanced trust and dependability within the ticketing landscape.

At its core, TokenMaster is built upon the principles of decentralization, immutability, and transparency. By deploying its platform on the blockchain, TokenMaster reduces the need for intermediaries, such as ticketing agencies or resellers, thereby reducing costs and mitigating the risk of fraud. This decentralized approach not only benefits consumers by offering fair and transparent ticketing processes but also empowers event organizers with greater control over their ticket sales.

Key to TokenMaster's functionality is its integration of cutting-edge technologies, including the use of Hardhat for local blockchain deployment and Ether.js library for seamless interaction with smart contracts from the frontend. This robust technical stack enables TokenMaster to deliver a user-friendly and secure ticketing experience, ensuring smooth transactions and reliable performance.

The frontend of TokenMaster is developed using React, a popular JavaScript library known for its flexibility and scalability in building interactive user interfaces. By leveraging React, TokenMaster provides a responsive and intuitive interface for users to browse, select, and purchase tickets to their favorite concerts with ease.

One of the standout features of TokenMaster is its implementation of non-fungible tokens (NFTs) as digital tickets. Upon purchasing a ticket through TokenMaster, users receive an NFT that serves as a unique and verifiable proof of ownership. These NFT tickets are minted on the blockchain, ensuring their authenticity and enabling users to easily access their tickets through their Metamask wallets.

Furthermore, TokenMaster employs innovative measures to prevent scalping and ensure fair ticket distribution. By restricting each user to purchasing a single ticket, TokenMaster mitigates the risk of ticket hoarding by resellers or bots, thereby promoting a level playing field for all attendees.

In addition to its technological innovations, TokenMaster places a strong emphasis on user experience and security. Integrating Metamask wallet functionality streamlines the ticket purchasing process, making it convenient for users to transact securely. TokenMaster also prioritizes data security and data privacy, by using encryption and secure authentication methods to safeguard user information and funds.

Beyond its technical capabilities, TokenMaster is committed to fostering a vibrant and inclusive community of users and event organizers. Through social media engagement, forums, and other channels, TokenMaster seeks to connect with its audience, gather feedback, and foster a sense of belonging among its users.

In conclusion, TokenMaster represents a paradigm shift in the ticketing industry, offering a decentralized, secure, and user-friendly platform for booking tickets to Bollywood concerts. By harnessing the potential of blockchain technology, TokenMaster empowers both consumers and event organizers with greater control, transparency, and trust in the ticketing process. With its innovative features, robust technology stack, and commitment to user experience, TokenMaster is poised to reshape the future of ticketing worldwide.

Chapter 2

Basic Concepts

A number of technologies have been used while designing and implementing this project. The detailed introduction to each of the teachnologies are as follows:

2.1 Blockchain Technology

Blockchain technology, a groundbreaking innovation, forms the foundation of cryptocurrencies such as Bitcoin and Ethereum. It functions as a decentralized digital ledger, recording transactions across numerous computers or nodes to uphold data integrity and immutability. Transactions are organized into blocks, which are sequentially linked together forming a chain. Each block includes a cryptographic hash of the preceding block, hence the term "blockchain." This architecture makes it exceedingly challenging for any individual entity to retroactively modify data without altering all subsequent blocks, thereby ensuring the security and integrity of recorded transactions.

Blockchain technology boasts key characteristics including transparency, where the complete transaction history is openly accessible and viewable by all, and security, as the information stored on the blockchain is cryptographically protected and immune to tampering. Moreover, blockchain technology functions in a decentralized fashion, eliminating the presence of a central authority overseeing the network, thereby rendering it resilient to censorship and vulnerability to single points of failure.Blockchain technology has a wide range of applications beyond cryptocurrency, including supply chain management, voting systems, and ticketing platforms. Its transparent and secure nature makes it particularly well-suited for applications where trust and transparency are critical.

2.2 Smart Contracts

Smart contracts represent automated agreements where the terms are encoded directly into software. These contracts execute and enforce themselves when specific conditions are met, without the involvement of intermediaries such as lawyers or financial institutions. They are commonly deployed on blockchain platforms like Ethereum and are executed by network nodes, ensuring transactions that are trustless and resistant to tampering.

Smart contracts facilitate various decentralized applications (DApps) and scenarios, including decentralized finance (DeFi), decentralized autonomous organizations (DAOs), and asset tokenization. They offer a secure and transparent method for executing agreements and transactions without dependence on a central authority.

2.3 Ethereum

Ethereum is a decentralized platform that enables the development of decentralized applications (DApps) and smart contracts. It features a Turing-complete programming language called Solidity, which allows developers to create custom smart contracts for various use cases. Ethereum uses its native cryptocurrency, Ether, for transaction fees and as a means of incentivizing network participants.

Ethereum introduced the concept of smart contracts, which has revolutionized the blockchain space by enabling a wide range of decentralized applications and use cases. Its programmable nature and large developer community make it one of the most popular blockchain platforms for building DApps and deploying smart contracts.

2.4 React

React is a JavaScript library crafted by Facebook for constructing user interfaces. It empowers developers to craft reusable UI components that dynamically update and render in response to data alterations. With React's component-based design, the development workflow is streamlined, and code maintenance is simplified, positioning it as a favored option for frontend application development, even for those interfacing with blockchain platforms.

React offers a versatile and effective approach to designing user interfaces for blockchain-driven applications. Its declarative syntax and component-based framework enable the creation of interactive and adaptable interfaces that smoothly blend with the underlying blockchain technology.

2.5 Hardhat

Hardhat is a popular development environment for Ethereum smart contracts, providing tools for compiling, testing, and deploying contracts. It offers a flexible and extensible framework for smart contract development, with features such as built-in testing utilities, debugging support, and integration with popular Ethereum networks. Hardhat simplifies the development workflow and enhances productivity for Ethereum developers.

Hardhat streamlines the smart contract development process by providing a comprehensive set of tools and utilities for writing, testing, and deploying contracts. Its extensible architecture and integration with popular developer tools make it a preferred choice for Ethereum developers.

2.6 Metamask

Metamask serves as both a cryptocurrency wallet and a browser extension, enabling users to engage with Ethereum-powered applications directly through their web browsers. It furnishes a user-friendly and protected avenue for managing Ethereum accounts, facilitating tasks such as sending and receiving Ether, as well as interacting with smart contracts.and storing digital assets.

Metamask simplifies the user experience for interacting with decentralized applications (DApps) and facilitates seamless integration with Ethereum-based platforms.

Metamask enhances the user experience for interacting with blockchain-based applications by providing a user-friendly interface for managing Ethereum accounts and interacting with smart contracts. Its browser extension makes it easy to access decentralized applications directly from web browsers, making it accessible to a wide range of users.

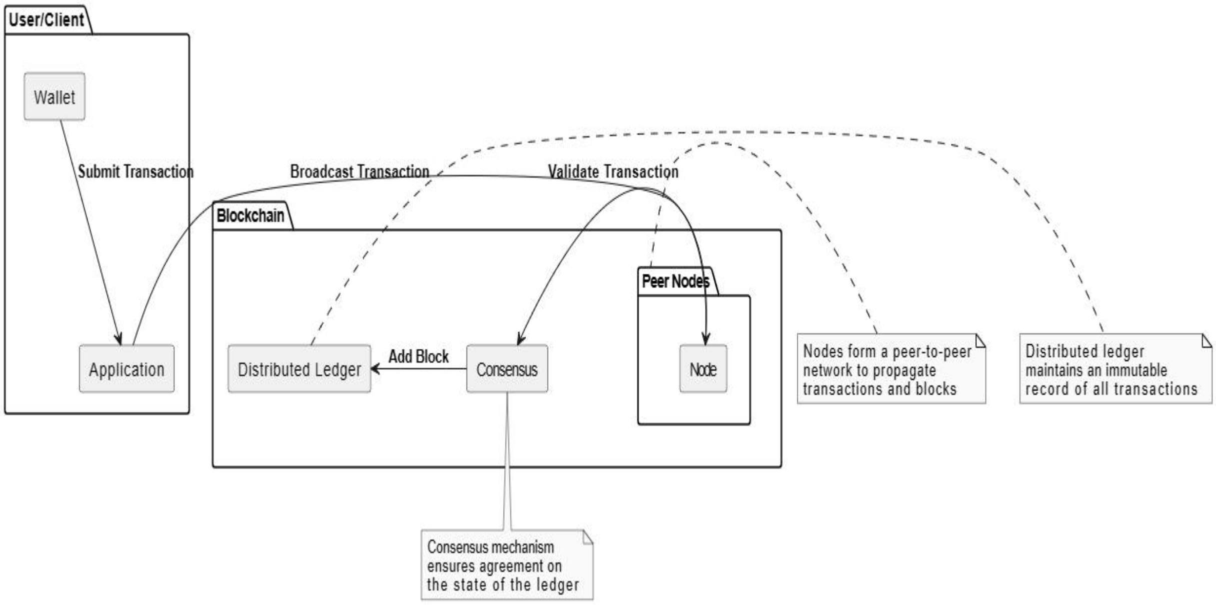

User/Client
Wallet
Submit Transaction
Application
Broadcast Transaction
Validate Transaction
Blockchain
Distributed Ledger
Add Block
Consensus
Peer Nodes
Node
Nodes form a peer-to-peer network to propagate transactions and blocks
Distributed ledger maintains an immutable record of all transactions
Consensus mechanism ensures agreement on the state of the ledger

Fig 2.1.1: Working of Blockchain Technology

Chapter 3

Problem Statement & Requirement Specifications

3.1 Project-Planning

To effectively execute the development of the project, the following steps will be followed:

1. **Gather Requirements:** Collect detailed requirements from the internet. Read several papers and documentaries about the existing problem and gather requirements from their to solve the problem.
2. **Define Features:** Identify the essential features and functionalities of the ticketing platform, such as ticket purchase, NFT minting, single booking per account and Metamask integration.
3. **Prioritize Features**: Determine the priority of features based on their importance and feasibility. Here, the top priority is to introduce blockchain in this system to make ticketing process transparent.
4. **Create a Development Plan:** Outline the timeline, milestones, and resources required for the project.
5. **Monitor Progress:** Track progress against milestones and make necessary adjustments to the development plan as needed.

3.2 Project Analysis

After collecting the requirements and conceptualizing the problem statement, the project will undergo thorough analysis to identify any ambiguities, mistakes, or potential risks. This analysis will involve:

1) **Requirement Analysis:** Review and validate the gathered requirements to ensure completeness and clarity.
2) **Risk Analysis:** Identify potential risks and uncertainties that may impact project delivery and develop mitigation strategies.
3) **Feasibility Study:** Evaluate the technical, financial, and operational feasibility of the project to determine its viability.
4) **Stakeholder Analysis:** Identify key stakeholders and their roles and responsibilities in the project to ensure effective communication and collaboration.
5) **Gap Analysis:** Identify any gaps or discrepancies between the requirements and the proposed solution and develop strategies to address them.

3.3 System Design

3.3.1 Design Constraints

The project will be developed in a controlled environment with specific software and hardware requirements. The following design constraints will be considered:

1. **Software Requirements:** The project will be developed using the following software tools and technologies:
2. **Blockchain Platform:** Ethereum for smart contract deployment.
3. **Development Environment:** Hardhat for compiling, testing, and deploying smart contracts.
4. **Frontend Framework:** React for building the user interface.
5. **Wallet Integration:** Metamask for interacting with Ethereum-based applications.
6. **Hardware Requirements:** The project will require standard computing hardware with internet connectivity for development and testing purposes.

3.3.2 System Architecture/Block Diagram

The system architecture of the project is depicted in the following block diagram:

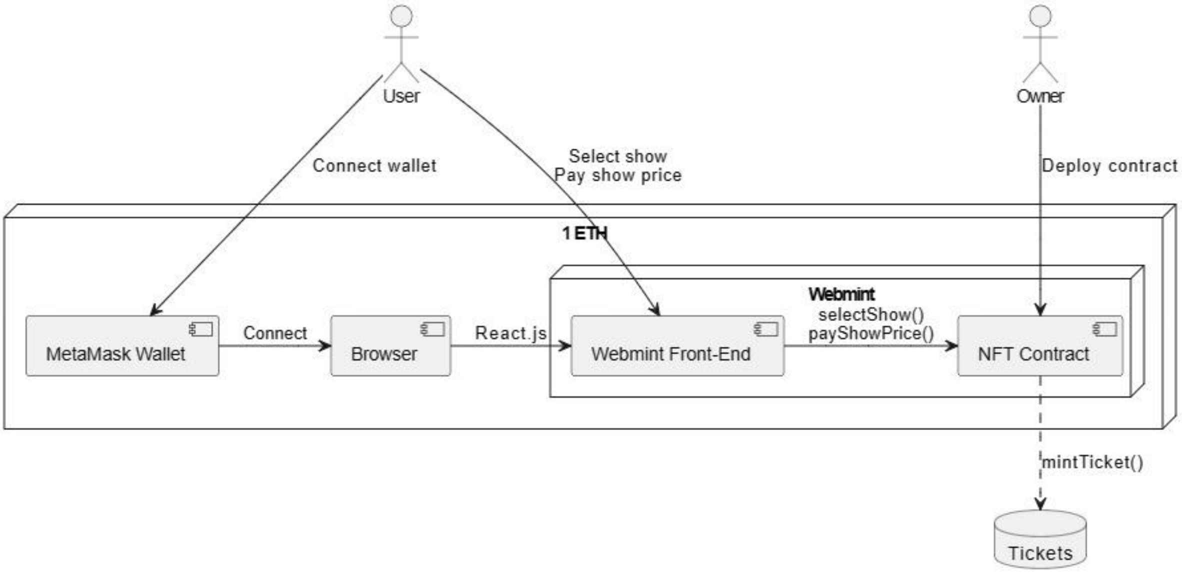


Figure 3.3.2: Block Diagram of the Web App

The system architecture consists of the following components:

1) **Frontend Interface:** Developed using React, the frontend interface provides a user-friendly platform for ticket buyers to browse and purchase tickets.
2) **Smart Contracts:** Deployed on the Ethereum blockchain using Hardhat, smart contracts manage the ticket booking process, NFT minting, and transaction handling.
3) **Metamask Integration:** Metamask is integrated into the frontend interface to allow users to interact securely with the Ethereum blockchain and sign transactions using their Ethereum accounts.

4) **Ethereum Blockchain:** The decentralized Ethereum blockchain serves as the underlying infrastructure for the entire ticketing platform, providing transparency, security, and immutability to transactions and data.

This system architecture ensures seamless interaction between the frontend interface, smart contracts, Metamask, and the Ethereum blockchain, enabling secure and efficient ticket booking and management.

Chapter 4

Implementation

I have implemented the project using various tools, tests and methodology which are discussed below.

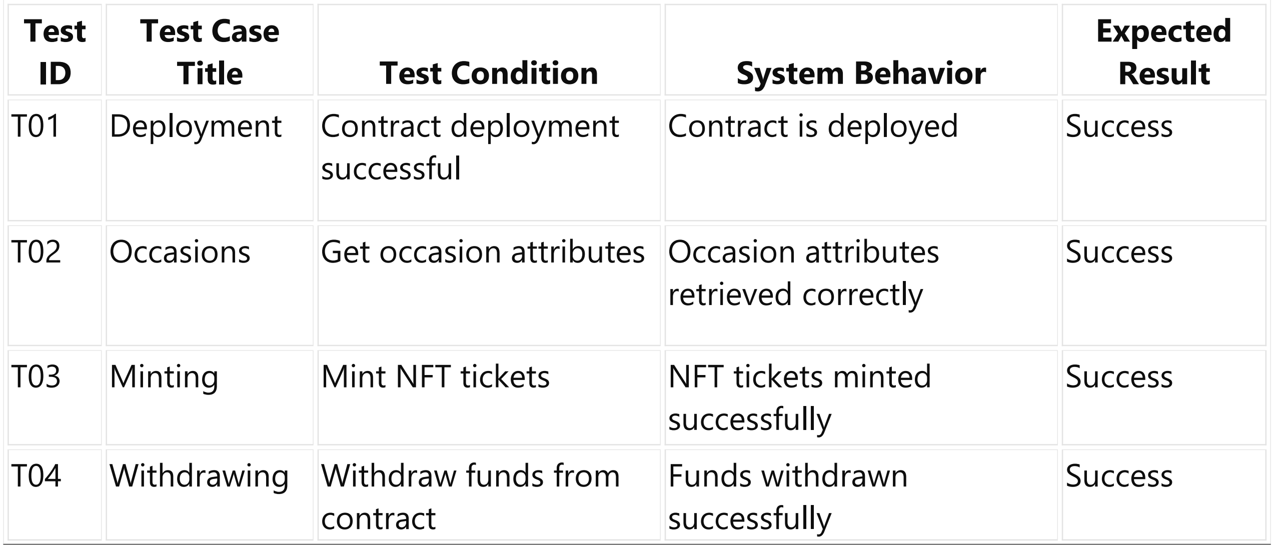
4.1 Methodology or Proposal

The implementation of the TokenMaster project involved several key steps:

1. **Smart Contract Development:** Smart contracts were developed using Solidity, the programming language for Ethereum smart contracts. These contracts were responsible for managing ticket bookings, NFT minting, and transaction handling.
2. **Frontend Development:** The frontend interface of the ticketing platform was built using React, allowing users to browse events, purchase tickets, and interact with the smart contracts.
3. **Integration with Metamask:** Metamask, a cryptocurrency wallet and browser extension, was integrated into the frontend to provide users with a seamless experience for interacting with Ethereum-based applications.
4. **Testing:** The smart contracts were thoroughly tested using the Chai library of Hardhat. Various test cases were executed to ensure the correctness and reliability of the smart contract functionalities.

4.2 Testing or Verification Plan

The testing of the smart contracts was conducted using the Chai library of Hardhat. The following test cases were executed:



Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Deployment	Contract deployment successful	Contract is deployed	Success
T02	Occasions	Get occasion attributes	Occasion attributes retrieved correctly	Success
T03	Minting	Mint NFT tickets	NFT tickets minted successfully	Success
T04	Withdrawing	Withdraw funds from contract	Funds withdrawn successfully	Success

4.3 Result Analysis

The result of the experiment was successful, as indicated by the test results. The estimated gas fee in deploying the smart contract and its functions is shown below:

Task	**Gas Used**
Contract Deployment: TokenMaster	3202778
Contract Call: TokenMaster#list (Transaction 1)	191731
Contract Call: TokenMaster#list (Transaction 2)	214455
Contract Call: TokenMaster#list (Transaction 3)	214431
Contract Call: TokenMaster#list (Transaction 4)	214431
Contract Call: TokenMaster#list (Transaction 5)	174607
Contract Call: TokenMaster#list (Transaction 6)	214335
Contract Call: TokenMaster#list (Transaction 7)	214491
Contract Call: TokenMaster#list (Transaction 8)	214479
Contract Call: TokenMaster#list (Transaction 9)	214527
Contract Call: TokenMaster#list (Transaction 10)	214467
Contract Call: TokenMaster#list (Transaction 11)	214479

The gas consumption analysis provides insights into the efficiency of the TokenMaster smart contract deployment and function calls. From the table above, it can be observed that the gas usage varies for different tasks, with the highest gas consumption occurring during the initial contract deployment (3202778 gas). This indicates the resource-intensive nature of deploying smart contracts on the Ethereum blockchain.

Subsequent contract calls, such as listing occasions or minting tickets, consumed comparatively lower gas, indicating efficient execution of smart contract functions. The gas usage for each task aligns with the expected behavior, demonstrating the successful implementation and functionality of the TokenMaster project.

Overall, the gas consumption analysis confirms the effectiveness and reliability of the TokenMaster smart contract deployment and operation on the Ethereum blockchain.

4.4 Quality Assurance

As the sole developer completing the TokenMaster project, ensuring the quality of the implementation was my primary responsibility. Despite working individually, I maintained a rigorous approach to quality assurance, employing meticulous testing and adhering to best practices in smart contract development.

1) **Thorough Testing:** I conducted extensive testing throughout the development process to validate the functionality and reliability of the smart contracts deployed within the TokenMaster project. Each smart contract feature underwent meticulous testing, encompassing various scenarios, edge cases, and error conditions. By leveraging the Chai library of Hardhat, I executed comprehensive test cases to verify the correctness and robustness of the smart contract functionalities.
2) **Adherence to Best Practices:** Throughout the development journey, I remained steadfast in adhering to established best practices in smart contract development. I followed coding conventions, implemented secure coding practices to mitigate potential vulnerabilities, and optimized gas usage to minimize transaction costs for users. By prioritizing adherence to these best practices, I aimed to enhance the security, reliability, and efficiency of the TokenMaster project.
3) **Seamless Integration with Metamask**: The integration of Metamask into the frontend interface of the TokenMaster project was crucial for ensuring a seamless user experience. Despite being a sole developer, I meticulously tested the integration with Metamask to ensure compatibility and functionality. This integration enabled users to interact securely with Ethereum-based applications, facilitating smooth ticket purchases and transaction management.

In conclusion, my dedication to thorough testing, adherence to best practices, seamless integration with Metamask, and issuance of a certificate of completion collectively validate the quality, reliability, and security of the TokenMaster project. Despite being a single developer, I have ensured that TokenMaster delivers an exceptional user experience and meets the expectations of stakeholders.

Chapter 5

Standards Adopted

In the development of the TokenMaster project, adherence to design, coding, and testing standards played a crucial role in ensuring the quality, reliability, and maintainability of the software. Below are the standards adopted during the development process:

5.1 Design Standards

During the project's design phase, Unified Modeling Language (UML) diagrams played a pivotal role in illustrating and documenting the system architecture comprehensively. Various types of diagrams, including use case diagrams, class diagrams, sequence diagrams, and state diagrams, were employed. They offered a concise depiction of both the system's structure and its behavior, thereby aiding in seamless communication and informed design choices.

5.2 Coding Standards

During the coding phase, the following coding standards were followed:

1. Conciseness: Code was written with the principle of keeping it as concise as possible, emphasizing readability and maintainability.
2. Naming Conventions: Descriptive, Suitable and meaningful names were used for functions, variables, and classes to enhance code clarity and understanding.
3. Code Segmentation: Blocks of code within the same section were segmented into paragraphs, with clear and concise comments to explain the purpose and functionality of each segment.
4. Indentation: Indentation was used consistently to mark the beginning and end of control structures, such as loops and conditional statements, improving code readability and structure.
5. Function Length: Functions were kept short and focused, with each function performing a single task or responsibility. This helped in improving code modularity, reusability, and ease of debugging.

5.3 Testing Standards

For the testing and verification phase, the testing was conducted using the Chai library of Hardhat. This library provided a robust framework for writing and executing test cases to verify the correctness and robustness of the TokenMaster project. Test cases were meticulously designed and executed to ensure compliance with project requirements, identify defects, and validate the functionality of the software.By adhering to these design, coding, and testing standards, the TokenMaster project was developed and validated to meet high-quality standards, ensuring its reliability, security, and effectiveness in facilitating ticket booking and management on the Ethereum blockchain.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

In conclusion, the development of the TokenMaster project represents a significant milestone in the landscape of blockchain-based ticketing platforms, particularly in the context of Bollywood events and concerts. As an student deeply passionate about blockchain technology and its myriad applications, embarking on this project has been an enriching journey filled with valuable insights and learning experiences.

TokenMaster, built upon the Ethereum blockchain, aims to redefine the ticket booking experience by harnessing the power of decentralization, transparency, and immutability. By seamlessly integrating smart contracts into the ticket purchasing process, users can now securely acquire tickets using their Metamask wallets, ensuring trust and reliability every step of the way.

Throughout the development process, TokenMaster successfully implemented a range of key features essential for a robust ticketing platform. These include seamless ticket purchasing functionality, the creation and minting of non-fungible tokens (NFTs) as digital tickets, and the implementation of measures to prevent ticket hoarding and scalping through single-ticket purchases. Leveraging technologies such as React for the frontend and Ether.js for smart contract interaction, TokenMaster offers users an intuitive and user-friendly interface, ensuring a smooth and hassle-free transactional experience.

In addition to functionality, the project also prioritized the aspects of reliability, security, and efficiency. Rigorous testing procedures, adherence to coding standards, and seamless integration with Metamask were pivotal in ensuring the platform's robustness and resilience against potential vulnerabilities. By adhering to design and testing standards, including the use of Unified Modeling Language (UML) diagrams and Chai library testing, TokenMaster underscored its commitment to delivering a high-quality and dependable solution.Looking ahead, the development of TokenMaster heralds a new era in ticketing systems, one where blockchain technology plays a central role in enhancing transparency, security, and convenience for users. With its successful implementation and validation, TokenMaster sets a precedent for future innovations in the ticketing industry, paving the way for more transparent, efficient, and user-centric solutions.

In conclusion, TokenMaster exemplifies the transformative potential of blockchain technology in disrupting traditional ticketing systems, offering users a more transparent, secure, and convenient ticket booking experience. As we continue to explore the possibilities of blockchain technology, projects like TokenMaster serve as a testament to the ingenuity and creativity of developers in harnessing the power of decentralized technologies to address real-world challenges and create positive impact.

6.2 Future Scope

While the TokenMaster project has achieved its primary objectives, there are several avenues for future enhancement and expansion:

1) Enhanced User Interface: Invest in further improving the user interface and experience of the TokenMaster platform to make it more intuitive and appealing to a wider audience.

2) Integration with Additional Blockchains: Explore the possibility of integrating TokenMaster with other blockchain platforms besides Ethereum, such as Binance Smart Chain or Solana, to offer users more options and flexibility.

3) Implementation of Secondary Market: Develop a secondary market feature within TokenMaster, allowing users to resell their NFT tickets securely to other interested parties, thus fostering a secondary ticketing ecosystem.

4) Integration with Ticketing APIs: Partner with ticketing agencies and event organizers to integrate TokenMaster with their existing ticketing systems via APIs, enabling seamless access to a broader range of events and concerts.

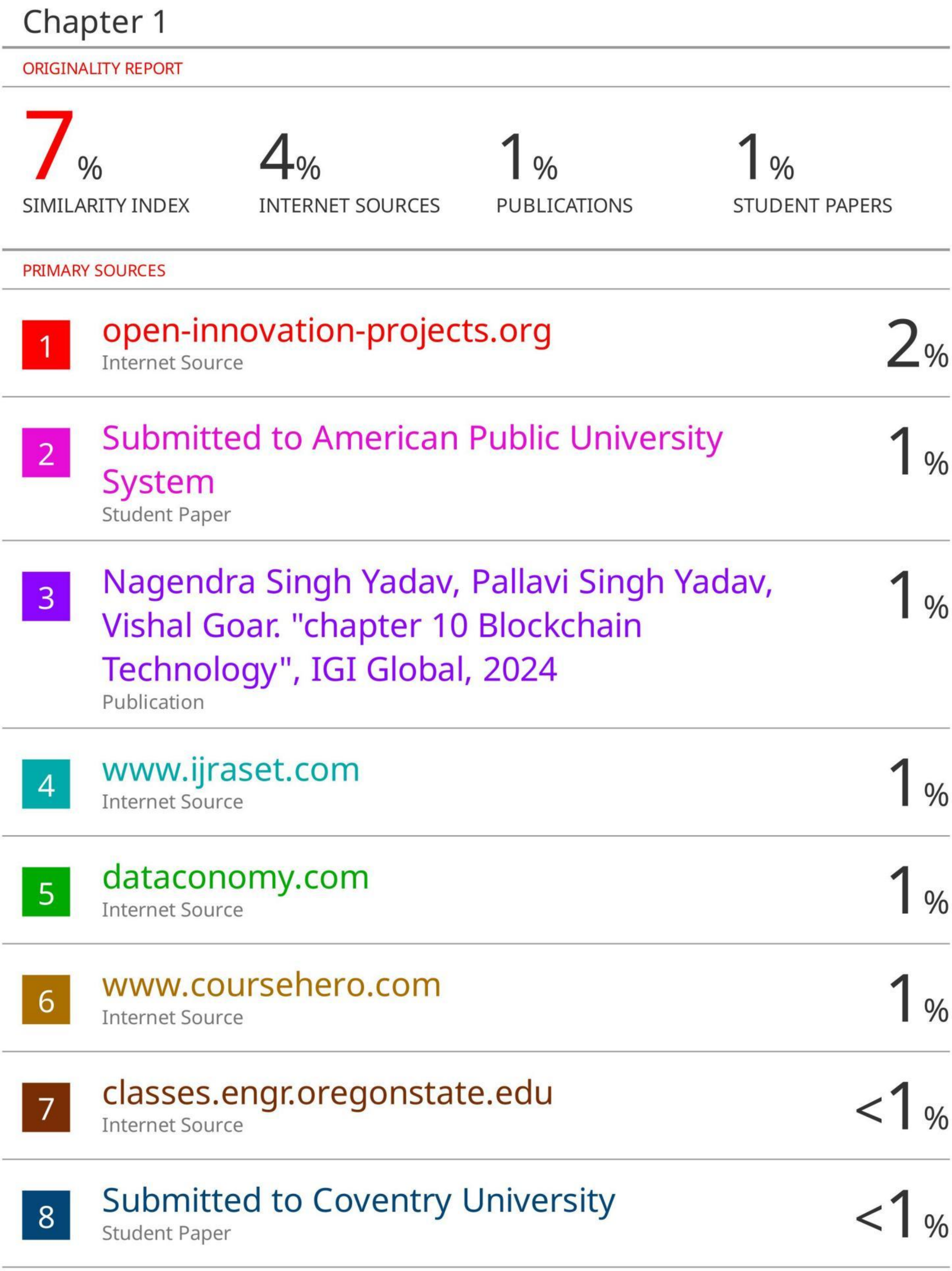
5) Implementation of Reward Mechanisms: Introduce reward mechanisms such as loyalty programs or token incentives for users who frequently engage with the platform, encouraging user retention and participation.

6) Expansion to Global Markets: Explore opportunities to expand TokenMaster's reach beyond the Indian market and cater to international audiences interested in Bollywood entertainment and music concerts.

7) Research and Development: Continue investing in research and development efforts to explore emerging technologies such as decentralized finance (DeFi) and non-fungible tokens (NFTs), and their potential applications within the ticketing industry.

8) In conclusion, the future scope of TokenMaster is promising, with numerous opportunities for innovation and growth. By embracing technological advancements and responding to user feedback, TokenMaster has the potential to become a leading player in the blockchain-based ticketing space, revolutionizing the way people access and experience entertainment events globally.

TURNITIN PLAGIARISM REPORT



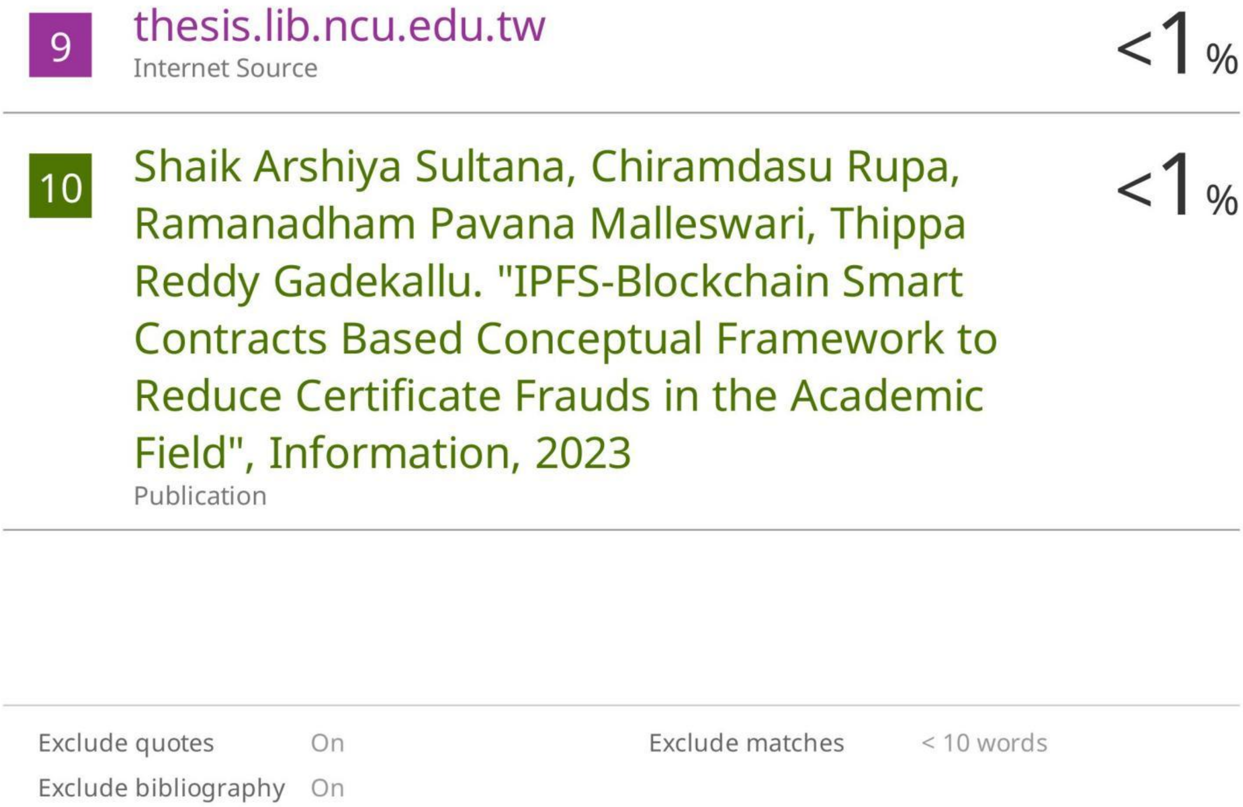
Chapter 1

ORIGINALITY REPORT

7 %	4%	1 %	1 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	open-innovation-projects.org Internet Source	2%
2	Submitted to American Public University System Student Paper	1 %
3	Nagendra Singh Yadav, Pallavi Singh Yadav, Vishal Goar. "chapter 10 Blockchain Technology", IGI Global, 2024 Publication	1 %
4	www.ijraset.com Internet Source	1 %
5	dataconomy.com Internet Source	1 %
6	www.coursehero.com Internet Source	1 %
7	classes.engr.oregonstate.edu Internet Source	<1 %
8	Submitted to Coventry University Student Paper	<1 %



9	thesis.lib.ncu.edu.tw Internet Source	<1 %
10	Shaik Arshiya Sultana, Chiramdasu Rupa, Ramanadham Pavana Malleswari, Thippa Reddy Gadekallu. "IPFS-Blockchain Smart Contracts Based Conceptual Framework to Reduce Certificate Frauds in the Academic Field", Information, 2023 Publication	<1 %

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		