

CRYPTISH LOTTERY DAPP

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

SAMIR KUMAR YADAV (Reg.No - 40111119)
SONU KUMAR SAW (Reg.No – 40111253)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
CATEGORY – 1 UNIVERSITY BY UGC

Accredited with Grade “A++” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Samir Kumar Yadav(40111119)** and **Sonu Kumar Saw(40111253)** who carried out the Project entitled " Cryptish-Lottery Dapp" under my supervision from November 2023 to April 2024 .

Internal Guide

Ms. YOGITHA.R M.E.,(Ph.D.,)

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

HEAD OF THE DEPARTMENT

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Jeyaraj Nagar, Rajiv Gandhi Salai,

Chennai - 600 119.

Submitted for Viva voce Examination held on

12/04/24

Internal Examiner

External Examiner

DECLARATION

Samir Kumar Yadav(Reg.No- 40111119), hereby declare that the Project Report entitled "Cryptish- lottery Dapp " done by me under the guidance of Ms. YOGITHA.R M.E.,(Ph.D.,) submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering .

DATE: 13-04-2024

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **The Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. YOGITHA.R M.E.,(Ph.D.,)** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Our lottery dapp is designed to allow users to purchase tickets using cryptocurrency and participate in a transparent and fair lottery. The dapp is built on the Ethereum blockchain and utilizes smart contracts to ensure the security and transparency of the lottery process. The front-end of the application is built using Next.js 13, which provides server-side rendering and automatic code splitting for improved performance and scalability. TailwindCSS is used for styling the application, which provides a range of pre-defined CSS classes to quickly and easily style the interface. The back-end of the application is built using Solidity, which is used to create a smart contract that manages the lottery process. The smart contract includes a function for buying tickets, which accepts cryptocurrency (MATIC) as payment and adds the user's address to the list of participants. Another function is used to randomly select a winner from the list of participants and distribute the winnings to the winner's address. This application is built using Remix IDE, Visual studio code editor, employing languages such as Javascript and solidity. It uses Ethereum blockchain network for storing and processing smart contract transactions.

TABLE OF CONTENT

CHAPTE R NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	1
2	LITERATURE SURVEY	2
3	REQUIREMENTS ANALYSIS	6
	3.1 Software Requirements Specification Document	6
4	DESCRIPTION OF PROPOSED SYSTEM	8
	4. 1 Architecture / Overall Design Of Proposed System	8
	4. 2 User Interface	10
	4. 3 Smart Contract Design	12
	4.4 Implementation Of Cryptish-lottery Dapp	14
5	IMPLEMENTATION DETAILS	15
	5.1 Smart Contract Architecture	15
	5.2 User Interface Design	15
	5.3 Integration with Web3.js	15
	5.4 Decentralized Lottery Creation	16
	5.5 Participant Interaction and Ticket Purchase	16
	5.6 Transparent Result Generation and Prize Distribution	16
	5.7 Security and Auditing Measures	16
	5.8 Continuous Improvement and Community Feedback	17
6	RESULTS AND DISCUSSIONS	18
	6.1 Platform Deployment and Accessibility	18
	6.2 User Interaction and Engagement	18

	6.3 Security and Transparency Measures	18
	6.4 Scalability and Performance Optimization	19
	6.5 Community Feedback and Iterative Development	19
	6.6 Future Directions and Challenges	19
7	CONCLUSION	23
	REFERENCES	24
	APPENDIX	25
	A. SOURCE CODE	25
	B. SCREENSHORTS	32
	C. RESEARCH PAPER	35

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
4.1	A Basic Lottery Dapp Architecture	8
4.2	Library Component using Metamask to Communicate with Ethereum	9
4.3	Login Page	10
4.4	Connecting Metamask	11
4.5	Admin was Logged in	11
4.6	Normal User Portal	12
4.7	Smart Contract Design	13
4.8	State Diagram	13
6.1	Connecting to Metamask	20
6.2	Player Enter into the Lottery	20
6.3	Winner Selected	21
6.4	Amount Transferred into the Winner's Account	21
6.5	Winner got the Amount	22
B.1	Metamsk Login	32
B.2	Remix IDE	33
B.3	NextJS	33
B.4	ReactJS	34
B.5	TailWindCSS	34

CHAPTER 1

INTRODUCTION

Decentralized applications (dapps) are becoming increasingly popular as they leverage blockchain technology to create platforms that are secure, transparent and decentralized. One such application is a lottery dapp, which aims to provide a transparent and secure platform for users to participate in lotteries. Unlike traditional lotteries that rely on a centralized authority to manage and conduct the lottery, lottery dapps utilize smart contracts to ensure fairness and transparency in the selection of lottery winners.

Our project is a lottery dapp that is built on the Polygon which is a layer 2 Ethereum blockchain using the Solidity programming language. The dapp allows users to purchase lottery tickets using cryptocurrency and automatically selects a winner based on a random algorithm. The winner is selected in a transparent and fair manner, ensuring that the lottery is free from any manipulation or fraud.

Our project aims to demonstrate the potential of blockchain technology in creating a secure and fair platform for lotteries. The use of smart contracts ensures that the lottery is conducted in a transparent and secure manner, which is free from any human intervention or bias. The platform also eliminates the need for intermediaries, such as banks or financial institutions, thereby reducing transaction costs and improving efficiency.

In this report, we provide a detailed description of the design and implementation of our lottery dapp. We also evaluate the functionality of the dapp and discuss any issues or challenges encountered during development. The report concludes with a discussion of the potential applications of lottery dapps and their impact on the blockchain industry.

CHAPTER 2

LITERATURE SURVEY

[1] This paper has handed me perceptivity about what guarantees does the integration of blockchain and power keep for electricity future. further, more i have found out about how structure plays an vital element in efficient price and we've got set this as our design goal.

[2] This paper stresses quantity decentralization and pseudonymity of the blockchain and has counselled approximately realistic functional and trouble model must be approached as their component is to sort exaggerate the miscalculations and faults inside the design in this case our layout on Lottery using blockchain.

[3] An assault referred to as document denial of chain which employs in a fraudulent way that the blockchain armature and it is settlement mechanisms to be able to control the stored comfy facts by rejecting licit changes or introducing unilateral mechanisms. And we are fastening our interest into enjoin this fault in our layout.

[4] This paper is possibly the primary paper to explain the transactional expenses of implementing blockchain technologies for incipency backing and. putting forward the limitations of blockchain because it does now not use introductory accept ed protocols that made the net extensively reachable like TCP -IP.

[5A] This paper has proven benefits and issues unique Coin presenting [ICO] and original alternate imparting [IEO] via a comparison with fairness conducting the lotteries and we've got allowed about using IOC&IEO as implicit pillars of our layout. A new model in the form of a decentralized app turned into designed inside the Ethereum blockchain to break the challenges gift and optimize the method of Zakat donation. Ethereum operates in an asynchronous terrain, the net. The command used to submit the sale via the JavaScript Console inside the dapp and Geth,web3.eth.send Transaction is with a message and that is also asynchronous in nature. hence, an asynchronous sub rate of deals offers bettered outturn.

[5B] The paper discusses about the anomalies incurring all through offline transaction in blockchain and the way they're reused after connecting to net and testing device of testnets to pretend the EVM and Ethereum community. and we are

the u sage of

outturn comparison of coetaneous, asynchronous and offline sale performed to load and strain testing.

[6] This paper has exceeded me with perceptivity connection between technologies and Financing. There are colourful orders of blockchain grounded on differing types. Blockc hains may be permissionless, where participation in the network settlement manner is open to any existent. For policymakers, this quandary is refocused as one it truly is at threat of failing to foster invention thru development, we will try to avoid this dilemma of chance failing.

[17] because blockchain supports a peer - to- peer structure, enables translucency, provenance , and invariability. Peer - to - peer shape is used for the storehouse of facts in every knot. they've a extra allotted records storehous e which facilitates to store storehouse area. This gets rid of the difficulty of having to host data in massive waiters in a single role. nonetheless, certainly with the 3 algorithms with the loftiest sequestration and security measures, it is continually visible that vicious bumps will always be in truth inside the block chain network. we are able to follow this version the use of smart contracts which solves the authentication trouble. This model provides the sequestration of the specification of a clever settlement, the sequestration of the prosecution o f a clever settlement, and the sequestration of the state of a smart agreement.

[18] The decentralization, enforceability, and verifiability traits of smart contracts allow agreement terms to be achieved among untrusted events with out the involvement of a trusted authority or a vital garçon. clever contracts are predicted to revise numerous conventional diligence, similar as financial, operation, IoT, and so forth. nonetheless, there are numerous boundaries additionally which encompass unrecoverable insect s, overall performance issues, lack of depended on data Feeds, lack of norms and rules.

[19] The possibility of operation Blockchain technology in colorful fields remains under take a look at. the decision for investor safety and protection in Lottery contracts may be spoke back by the preface of blockchain technology, which functions on a believe -free gadget where individualities have little to do to make it work. it's clean, comfy, and handy means for the exchange of data and the switch of price ran ge. The era is fluently programmable. however there may be a lack of mindfulness. there is loads of discussion approximately blockchain, but humans

don't know the genuine cost of blockchain and the way they could practice it in one - of -a kind conditions.

[10] the principle advantage of these structures is, human beings can make contributions plutocrat without any trustfactor. this is the place in which conventional crowdfunding platforms lacks. every other trouble of the being technique is the centralised wa iters, that are difficulty to vicious exertion.

[21] The layout is an internet operation which is basically an improvement of the being transparency in lottery systems. In ordinary Lottery as there's no safety assured for participate quantum. So, clever settlement of Ethereum platform that is an operation of blockchain can be used in order to interrupt this trouble. Blockchain is a decentralized all otted tally machine that accesses, verifies, and transmits community data via distributed bumps.

[22] From this paper what i have found out is that we are using Testnet and Meta masks. The experie nce used in this touch is reliability which has surpassed outcomes in which it's been examined on n Ganache unique community which enables for the speed of transactions. The perpetration of this concept contains loopholes which endure correction for the sa fety of the fund, we are able to try and discover the end result for those loopholes.

[13] smart contracts do no longer require brokers or other intermediaries to validate the agreement, so that they eliminate the chance of manipulation by using 1/3 events, ensuing in fee savings and reduced fraud. clever contracts are assigned, it's miles almost impossible to trade previously assigned ones, and loopholes are feasible. A blockchain -primarily based clever settlement requires a excessive aid fee. smart contracts use cryptocurrency, which now not all governments have legalized.

[14] The knowledge I have ate up in this work is that blockchain transactions and clever contracts may be used in lots of areas, starting from coverage claims to the economic quarter. For safety purposes, we may additionally upload patterns to explain positive attacks to assist us apprehend what wishes to be designed to assist the ones assaults.

[15] . We adopt blockchain technology and smart contract, as well as RANDAO [10] in the design of DeLottery. The whole system follows Initial Phase, Interaction Phase and Ending Phase, each including two detailed steps. The six steps are easy to implement by Solidity, and ensure efficiency, since each step has a time

complexity no more than

$O(n)$. This feature provides DeLottery with a wide range of applications, especially when dealing with huge amount of data.

CHAPTER 3

REQUIREMENT ANALYSIS

3. 1 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

3. 1.1 Solidity

Solidity is an object oriented high -level language used for writing smart contracts in the crowdfunding dapp. A statically typed language has features such as inheritance, libraries and complex user defined types. C++, python and JavaScript , majorly influenced it. Solidity is compiled to bytecode that is executable by Ethereum virtual machine (EVM). Using solidity developers can write dapps that implement self enforcing business logic contained in smart contracts, leaving an undeniable and permanent record of transactions .

3.1.2 Visual Studio

Visual studio code is a code editor developed by Microsoft. It is a feature rich editor that supports code highlighting, code debugging, intelligent code completion and code refactoring to build web applications. It supports wide range of programming languages and has a built in terminal to execute command line commands. As Lottery is a frontend, heavy application with lots of Javascript Visual studio code was a best choice as the editor to develop the application .

3.1.3 Metamask

Metamask is a chrome browser plugin that acts as a bridge between your browser and Ethereum blockchain by providing a secure identity vault, a user interface to manage multiple Ethereum wallets and sign blockchain transactions. It is one of the best ways to send transactions to Ethereum blockchain because it keeps a track of transaction execution and returns if any error occurs during mining or execution. It supports any ERC20 type token to be added to your wallet and trigger transaction on

those ERC20 tokens. It is an Ethereum community open source project having more than million active users; hence, it is the most popular plugin to interact with blockchain.

3.1.4 *Next.js*

It is a popular framework for building server-side rendered React applications. It provides several features that make development faster and more efficient, such as automatic code splitting, optimized performance, and easy deployment.

3.1.5 *Thirdweb*

It is a blockchain infrastructure provider that offers a variety of tools and services for building decentralized applications. It provides a seamless integration of blockchain technology with traditional web development, making it easy for developers to build and deploy dapps.

3.1.6 *Ether.js*

It is a JavaScript library for interacting with the Ethereum blockchain. It provides a simple and easy-to-use API for sending and receiving transactions, managing wallets, and interacting with smart contracts.

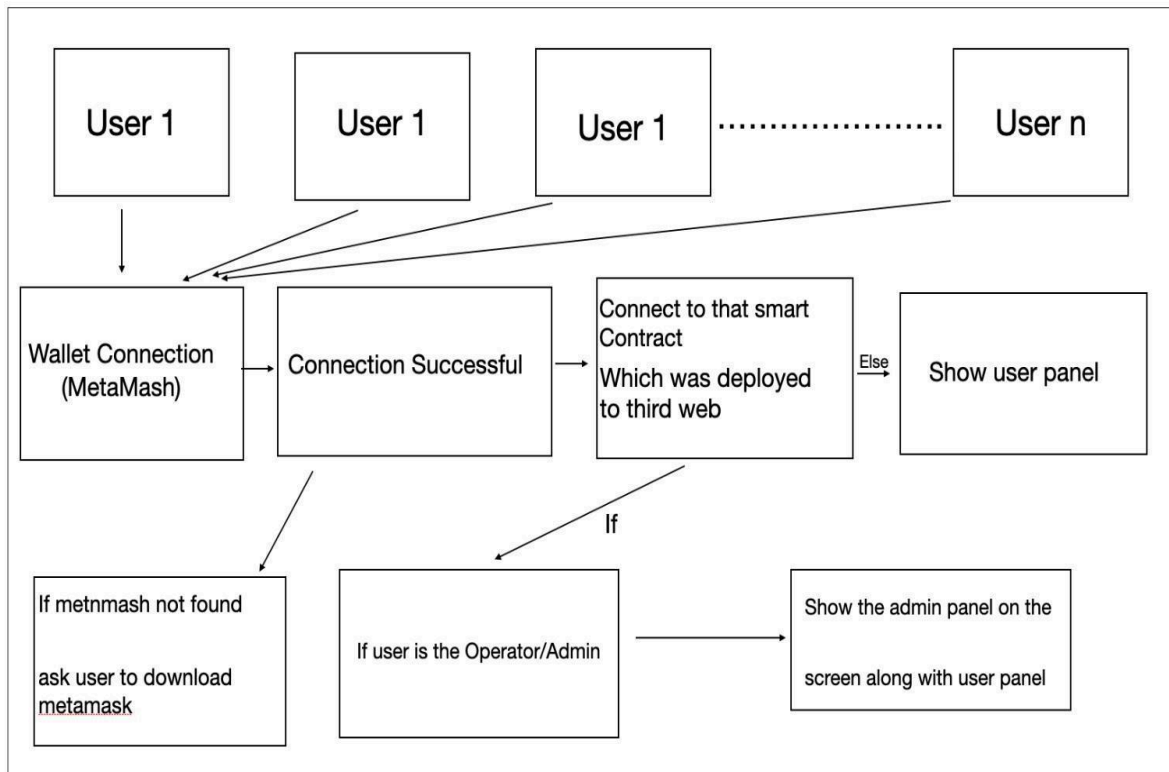
3.1.7 *TailwindCSS*

TailwindCSS is a utility-first CSS framework that provides a set of pre-defined CSS classes to enable rapid styling of web applications. It allows developers to quickly create custom designs using a range of pre-defined utility classes.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM



Fi g 4.1: A Basic Lottery Dapp Architecture

Our lottery dapp is designed to allow users to purchase tickets using cryptocurrency and participate in a transparent and fair lottery. The dapp is built on the Ethereum blockchain and utilizes smart contracts to ensure the security and transparency of the lottery process. The front-end of the application is built using Next.js 13, which provides server-side rendering and automatic code splitting for improved performance and scalability.

TailwindCSS is used for styling the application, which provides a range of pre-defined CSS classes to quickly and easily style the interface.

The back-end of the application is built using Solidity, which is used to create a smart contract that manages the lottery process. The smart contract includes a function for buying tickets, which accepts cryptocurrency (MATIC) as payment and adds the user's address to the list of participants. Another function is used to randomly select a winner from the list of participants and distribute the winnings to the winner's address.

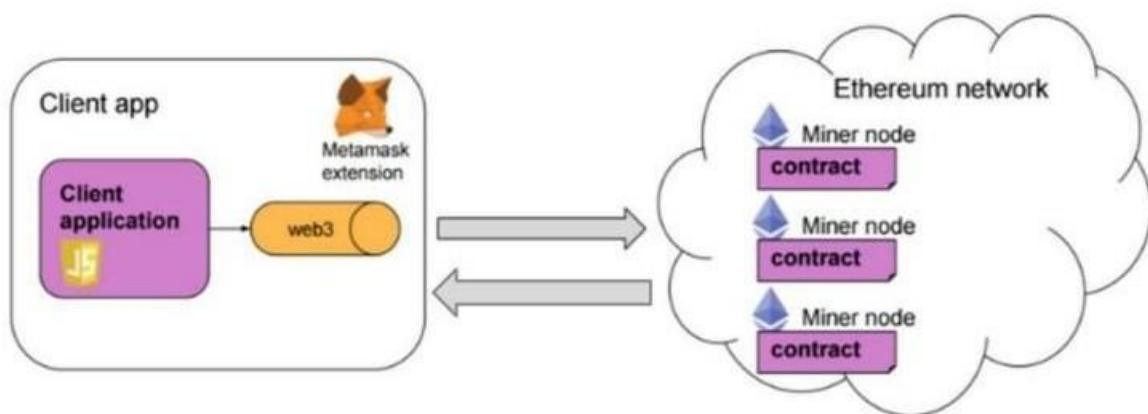


Fig 4.2: Library Component using Metamask to Communicate with Ethereum

As shown in Figure 4.2 library component using Metamask to communicate with Ethereum nodes library component in the dapp uses web3 (Ethereum JavaScript API) to trigger transactions depending on the user interaction with the view component. Metamask acts as lightweight client that provides async functionality to library component for making calls to Ethereum network using API's provided by web3. Metamask also tracks the transaction processing and updates the library component about the status of mining of that transaction. A transaction needs to be signed before it can be sent to Ethereum blockchain. Metamask handles this by using the API's provided in web3 library. Once the transaction is signed Metamask will make a RPC call to send the transaction to Ethereum network where the smart contracts are deployed. Metamask polls Ethereum network to check the status of the transaction and update the Library component using a callback function present in library component.

The other network calls handled by library component is to the IPFS to store and retrieve campaign metadata. IPFS provides standard set of API's to interact with its file system. Library component makes web service calls to Our lottery dapp is designed to allow users to purchase tickets using cryptocurrency and participate in a

transparent

and fair lottery. The dapp is built on the Ethereum blockchain and utilizes smart contracts to ensure the security and transparency of the lottery process. The front-end of the application is built using Next.js 13, which provides server-side rendering and automatic code splitting for improved performance and scalability.

TailwindCSS is used for styling the application, which provides a range of pre-defined CSS classes to quickly and easily style the interface.

The back-end of the application is built using Solidity, which is used to create a smart contract that manages the lottery process. The smart contract includes a function for buying tickets, which accepts cryptocurrency (MATIC) as payment and adds the user's address to the list of participants. Another function is used to randomly select a winner from the list of participants and distribute the winnings to the winner's address.

This application is built using Remix IDE, Visual studio code editor, employing languages such as Javascript and solidity. It uses Ethereum blockchain network for storing and processing smart contract transactions.

4.2 USER INTERFACE

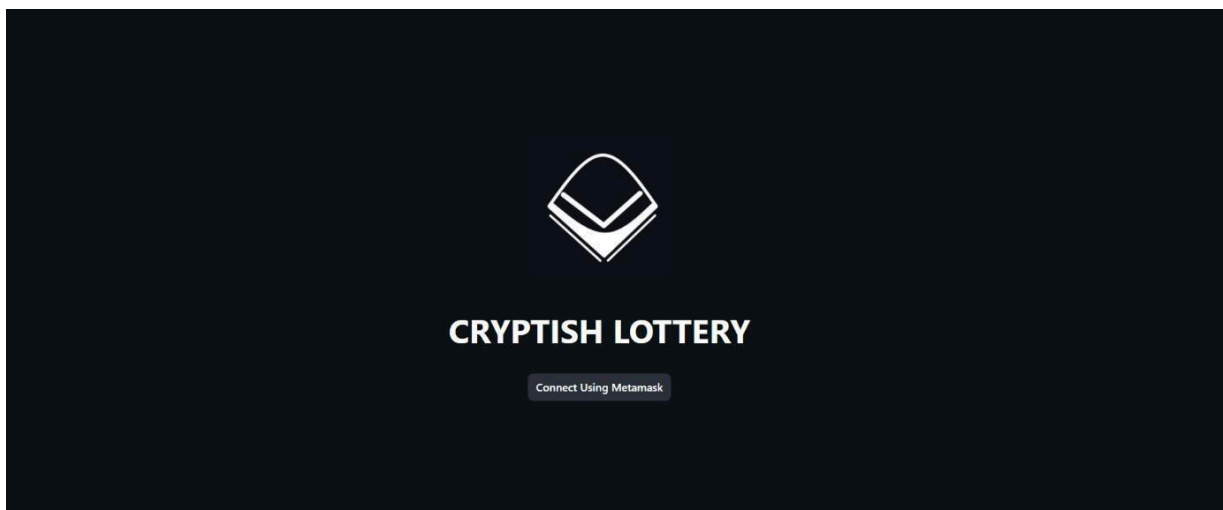


Fig 4.3: Login Page

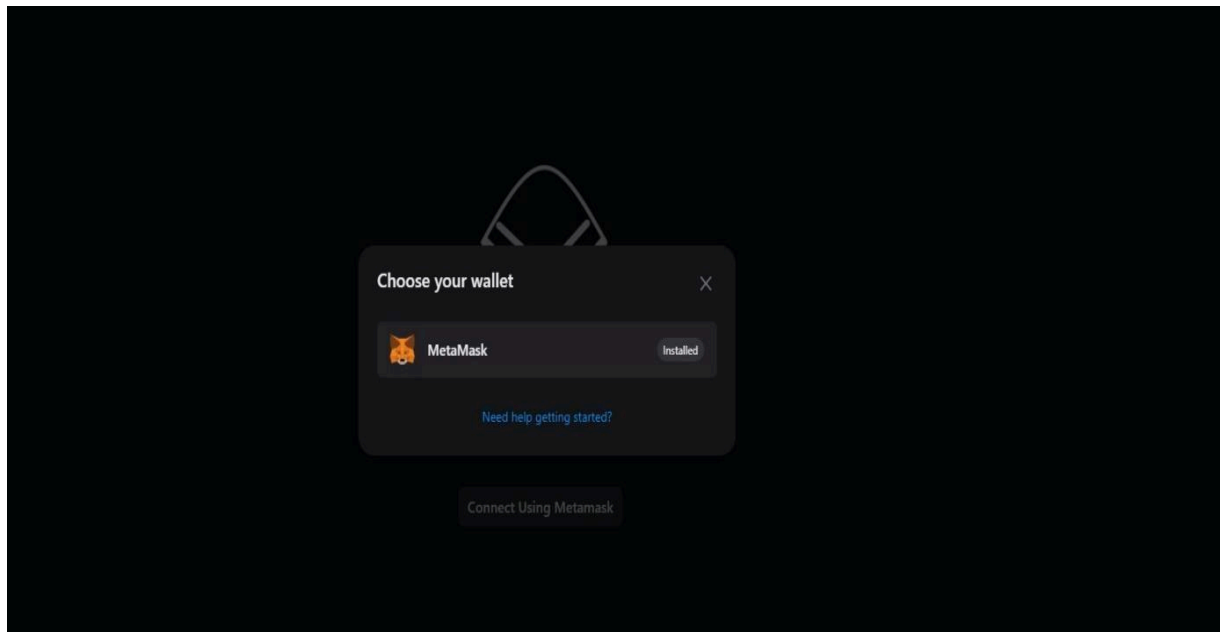


Fig 4.4 Connecting Metamask

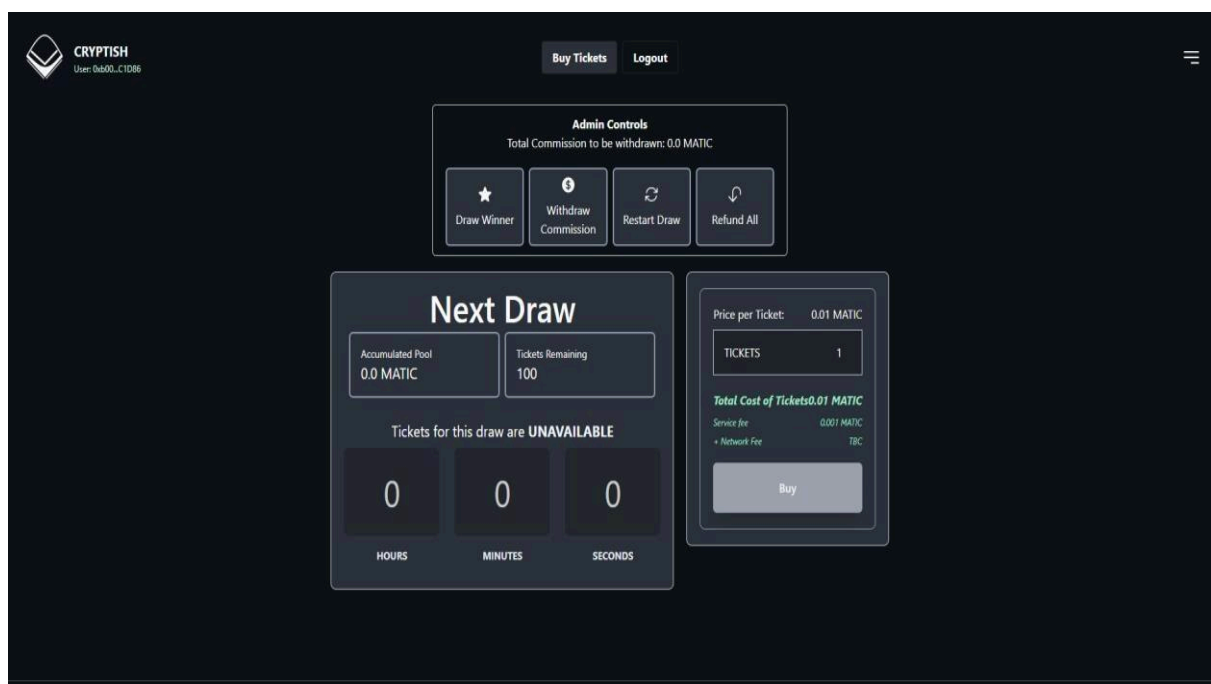


Fig 4.5: Admin was logged in

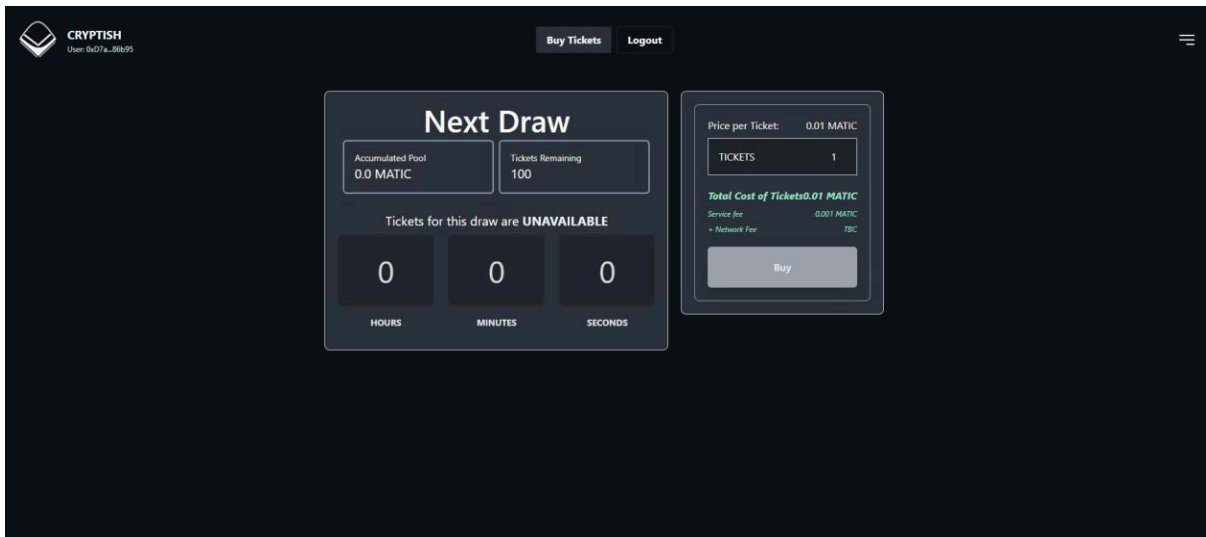


Fig 4.6: Normal user portal

When the user is asked to connect their metamask account, they are sharing their wallet address to the dapp and then in the NextJs code we check address is same as the address used to deploy the smart contract over to thirdweb and if yes then we show the admin panel otherwise the application only shows encapsulated panel for the user.

4.3 SMART CONTRACT DESIGN

The smart contracts written for Lottery dapp tie the whole platform together and make it work by controlling the flow of business logic. Even though they do not contain any of the bulk data, the business model they define and transactions that change the business logic flow are how this system of Lottery can work in a peer-to-peer manner, without central control.

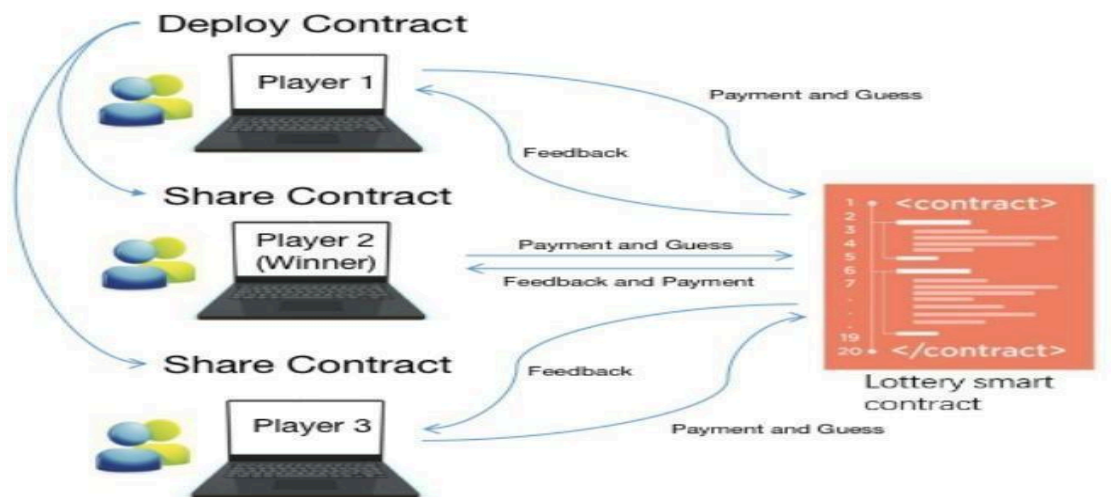


Fig 4 .7: Smart Contract Design



Fig 4 .8: State Diagram

The smart contract has several parameters set, including the price of each ticket, the maximum number of tickets available for purchase, and the commission fee charged per ticket. The lottery also has a set duration after which it expires if it is not carried out, and the operator of the lottery is able to withdraw commission fees earned from ticket sales. The smart contract includes various functions such as buying tickets, drawing a winner, restarting a draw, checking winnings, withdrawing winnings, refunding tickets, and withdrawing commissions. The smart contract also includes a function to check if the caller is a winner or the lottery operator.

4.4 IMPLEMENTATION OF LOTTERY DAPP

Our lottery dapp is designed to allow users to purchase tickets using cryptocurrency and participate in a transparent and fair lottery. The dapp is built on the Ethereum blockchain and utilizes smart contracts to ensure the security and transparency of the lottery process. The front-end of the application is built using Next.js 13, which provides server-side rendering and automatic code splitting for improved performance and scalability.

TailwindCSS is used for styling the application, which provides a range of pre-defined CSS classes to quickly and easily style the interface.

The back-end of the application is built using Solidity, which is used to create a smart contract that manages the lottery process. The smart contract includes a function for buying tickets, which accepts cryptocurrency (MATIC) as payment and adds the user's address to the list of participants. Another function is used to randomly select a winner from the list of participants and distribute the winnings to the winner's address.

Ethers.js is used to interact with the smart contract from the front-end of the application. It provides a simple and easy-to-use API for sending and receiving transactions, managing wallets, and interacting with smart contracts.

Metamask Wallet is used to provide users with a secure and convenient way to manage their cryptocurrency and participate in the lottery.

Thirdweb is used to deploy the smart contract to the Ethereum blockchain. It provides a range of tools and services for managing blockchain infrastructure and simplifies the process of deploying smart contracts to the blockchain.

During the implementation process, we encountered several challenges, including ensuring the security and integrity of the smart contract, implementing a random algorithm for selecting the winner, and integrating the smart contract with the front-end of the application. However, we were able to overcome these challenges through rigorous testing and collaboration between the front-end and back-end development teams.

Overall, our implementation has resulted in a secure and efficient lottery dapp that leverages the unique properties of blockchain technology to provide a transparent and fair platform for conducting lotteries. We believe that our project demonstrates the potential of blockchain technology in creating decentralized applications that are

free from intermediaries and manipulation.

CHAPTER 5

IMPLEMENTATION DETAILS

The implementation of the DeLottery platform encompasses various activities aimed at creating a seamless user experience while ensuring the security and reliability of the system. Leveraging a combination of smart contracts, frontend libraries, and decentralized technologies, the platform delivers a robust solution for hosting and participating in decentralized lotteries. This chapter delves into the intricacies of the implementation, highlighting key components and functionalities.

5.1 SMART CONTRACT ARCHITECTURE

The foundation of the DeLottery platform is built upon a series of smart contracts deployed on the Ethereum blockchain. These contracts govern essential functionalities such as lottery creation, participant interaction, and prize distribution. The contracts are designed using Solidity, Ethereum's programming language for smart contracts, and adhere to best practices for security and efficiency.

5.2 USER INTERFACE DESIGN

The user interface (UI) of the DeLottery platform is developed using modern frontend technologies, including HTML, CSS, and JavaScript libraries such as React.js. The UI is designed with a focus on simplicity, intuitiveness, and responsiveness, allowing users to easily navigate the platform and participate in lotteries without friction. Components such as lottery creation forms, participant dashboards, and result displays are meticulously crafted to provide a seamless user experience across devices.

5.3 INTEGRATION WITH WEB3.JS

To interact with the Ethereum blockchain and smart contracts, the DeLottery platform leverages Web3.js, a JavaScript library that enables communication with Ethereum nodes. Web3.js facilitates seamless integration between the frontend UI and blockchain backend, allowing users to perform actions such as lottery participation, fund transfers, and transaction monitoring directly from their browsers. Integration with Web3.js also enables real-time updates and event notifications, enhancing user engagement and transparency.

5.4 DECENTRALIZED LOTTERY CREATION

The process of creating a decentralized lottery on the DeLottery platform is facilitated through a user-friendly interface that guides lottery creators through the setup process step by step. Lottery creators provide essential details such as lottery duration, ticket price, prize distribution mechanism, and eligibility criteria using intuitive forms and dropdown menus. Once all parameters are defined, the lottery creation transaction is initiated, deploying a new smart contract instance on the blockchain.

5.5 PARTICIPANT INTERACTION AND TICKET PURCHASE

Participants interact with the DeLottery platform through a dedicated dashboard where they can browse active lotteries, view lottery details, and purchase tickets using cryptocurrency. The ticket purchase process is streamlined, with participants able to select the number of tickets desired and confirm the transaction using their connected Ethereum wallet, such as MetaMask or Trust Wallet. Real-time updates on ticket availability and lottery status ensure participants are informed throughout the process.

5.6 TRANSPARENT RESULT GENERATION AND PRIZE DISTRIBUTION

Upon the conclusion of a lottery, the DeLottery platform employs a transparent and verifiable process for result generation and prize distribution. A provably fair random number generation mechanism is utilized to select winning tickets, ensuring integrity and fairness in the outcome. Smart contracts automatically distribute prizes to

winning

participants' Ethereum addresses, with transaction details recorded on the blockchain for auditing and verification purposes.

5.7 SECURITY AND AUDITING MEASURES

Security is paramount in the implementation of the DeLottery platform, with robust measures in place to protect user funds and prevent malicious activities. Smart contracts undergo rigorous testing and auditing by security experts to identify and mitigate potential vulnerabilities. Standardized security practices, including code review, static analysis, and formal verification, are employed to ensure the integrity and reliability of the platform.

5.8 CONTINUOUS IMPROVEMENT AND COMMUNITY FEEDBACK

The development of the DeLottery platform follows an iterative process driven by community feedback and continuous improvement. Feedback mechanisms, such as user surveys, community forums, and social media channels, are utilized to gather insights and suggestions from platform users. Regular updates and feature enhancements are rolled out based on user feedback, ensuring the platform remains responsive to user needs and preferences.

CHAPTER 6

RESULTS AND DISCUSSIONS

In this chapter, we present the results of the DeLottery platform implementation and discuss its implications for decentralized lottery hosting. The platform's functionality, user experience, and security measures are evaluated, providing insights into its effectiveness and potential areas for improvement.

6.1 PLATFORM DEPLOYMENT AND ACCESSIBILITY

The deployment of the DeLottery platform marks a significant milestone in enabling decentralized lottery hosting accessible to a global audience. By leveraging blockchain technology and smart contracts, the platform eliminates geographical barriers and allows users from anywhere in the world to participate in lotteries securely and transparently. The accessibility of the platform is evidenced by the creation of multiple decentralized lotteries, each attracting participants from diverse backgrounds.

6.2 USER INTERACTION AND ENGAGEMENT

The user interaction and engagement on the DeLottery platform play a crucial role in its success and sustainability. Through intuitive user interfaces and seamless integration with Ethereum wallets, participants can easily navigate the platform, purchase lottery tickets, and monitor lottery results in real-time. The platform's user-friendly design fosters a positive user experience, encouraging continued participation and community growth.

6.3 SECURITY AND TRANSPARENCY MEASURES

Security and transparency are paramount in the context of decentralized lotteries, where trust and integrity are essential for user confidence. The implementation of

robust security measures, including smart contract auditing, encryption protocols,
and

multi-factor authentication, ensures the protection of user funds and sensitive data. Additionally, the transparency of lottery results and prize distribution mechanisms, facilitated by on-chain transaction records and cryptographic verification, instills trust in the platform among participants.

6.4 SCALABILITY AND PERFORMANCE OPTIMIZATION

As the DeLottery platform continues to attract a growing user base and host an increasing number of decentralized lotteries, scalability and performance optimization become imperative. Efforts to enhance platform scalability through layer 2 scaling solutions, such as sidechains and state channels, are essential to accommodate higher transaction volumes and maintain optimal user experience. Continuous monitoring and optimization of platform performance metrics, including transaction throughput and latency, ensure efficient operation and user satisfaction.

6.5 COMMUNITY FEEDBACK AND ITERATIVE DEVELOPMENT

Community feedback plays a vital role in shaping the evolution of the DeLottery platform, guiding iterative development and feature enhancements. Through community forums, user surveys, and social media channels, participants contribute valuable insights and suggestions for platform improvement. The integration of user feedback into development roadmaps ensures that the platform remains responsive to user needs and preferences, fostering a vibrant and engaged community of users and developers.

6.6 FUTURE DIRECTIONS AND CHALLENGES

Looking ahead, the DeLottery platform faces both opportunities and challenges in its quest to revolutionize decentralized lottery hosting. Key areas of focus include further scalability improvements, regulatory compliance, and integration with emerging technologies such as decentralized finance (DeFi) and non-fungible tokens (NFTs). Addressing regulatory considerations and compliance requirements while preserving the platform's decentralized ethos presents a significant challenge.

However, with

proactive community engagement and collaborative development efforts, the platform is well-positioned to overcome challenges and drive innovation in the decentralized lottery space.

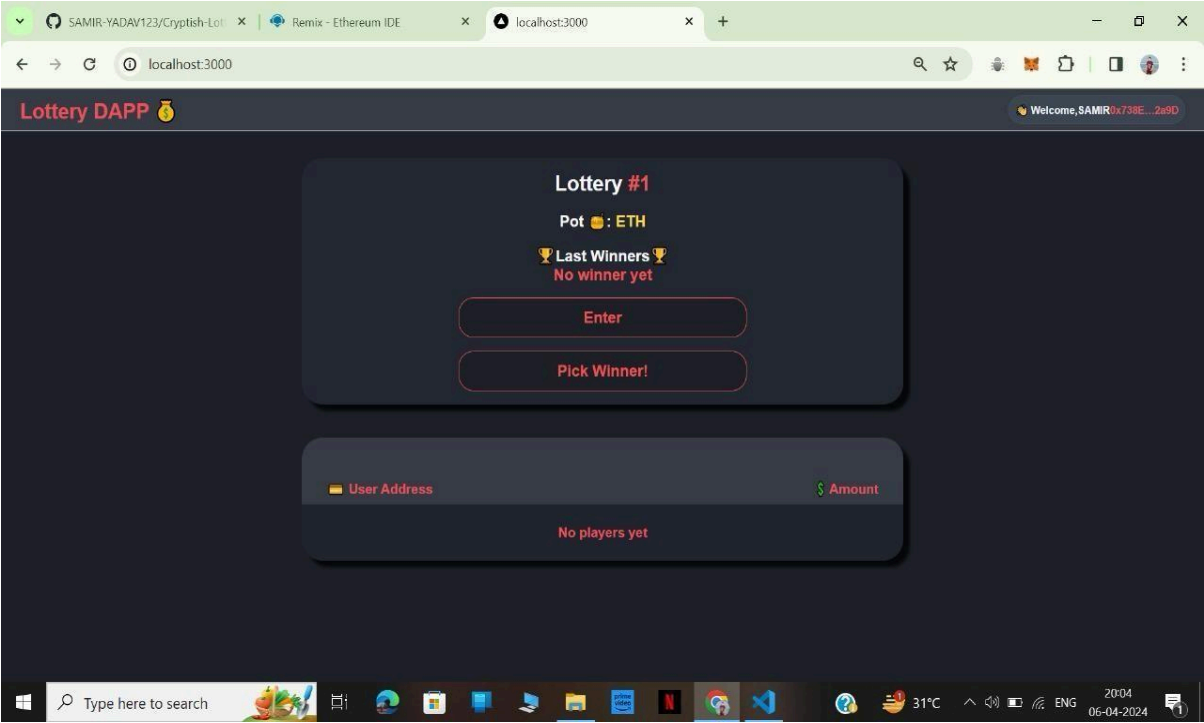


Fig 6.1: Connecting to Metamsk

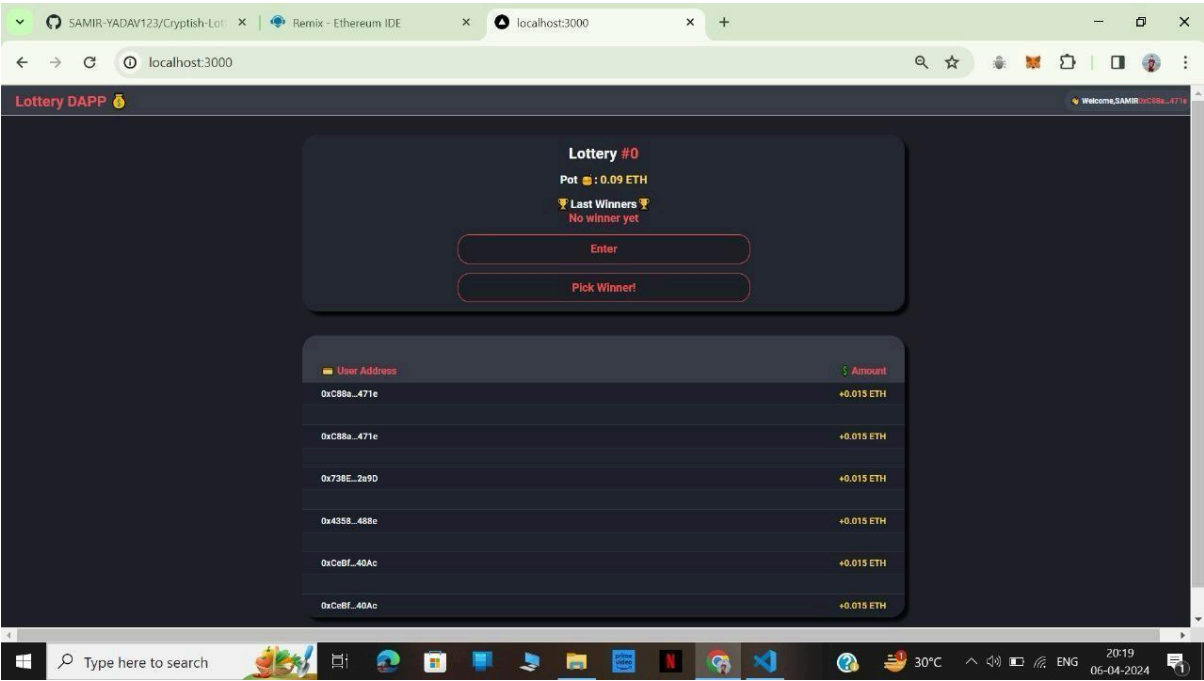


Fig 6.2: Player Enters into the Lottery

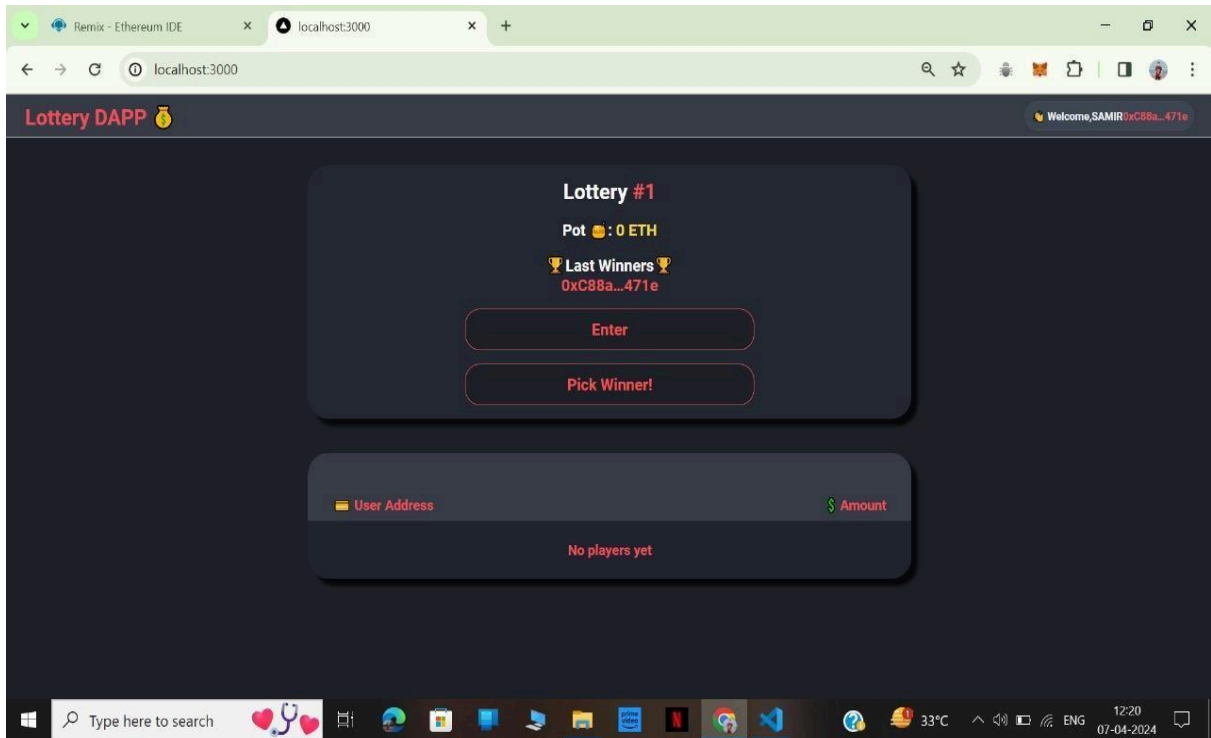


Fig 6.3: Winner Selected

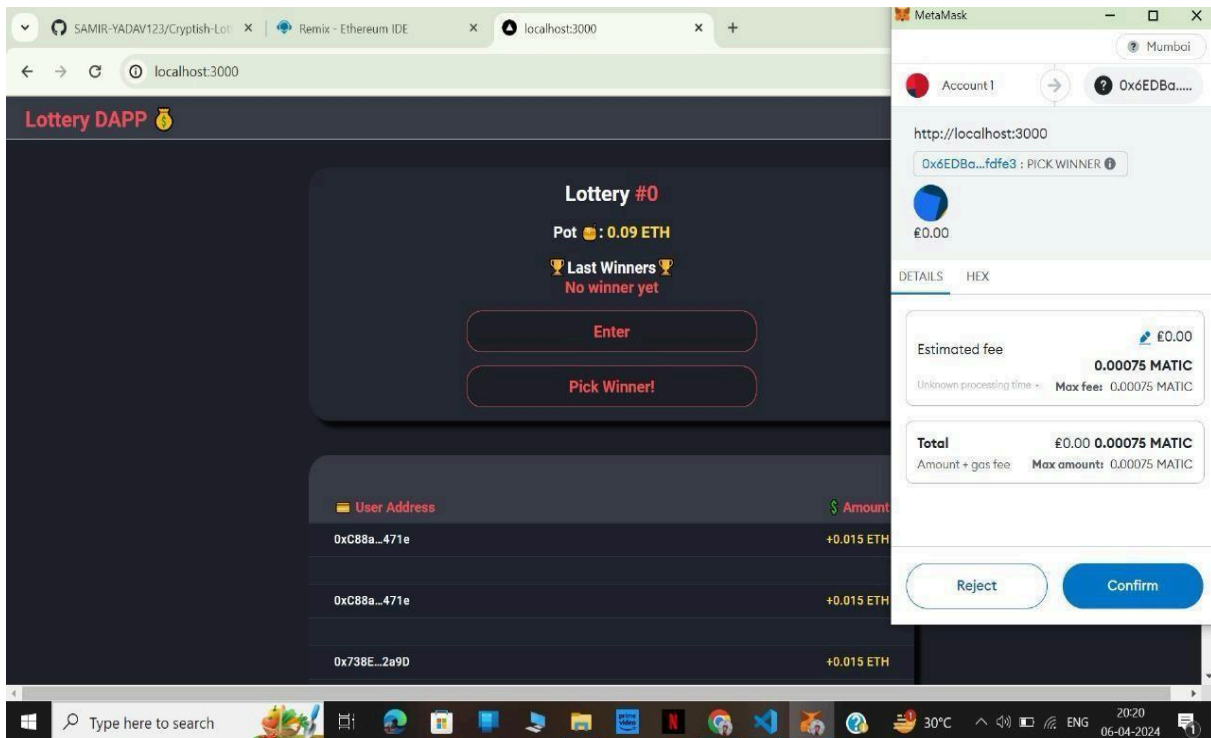


Fig 6.4: Amount Transferred into the Winner's Account

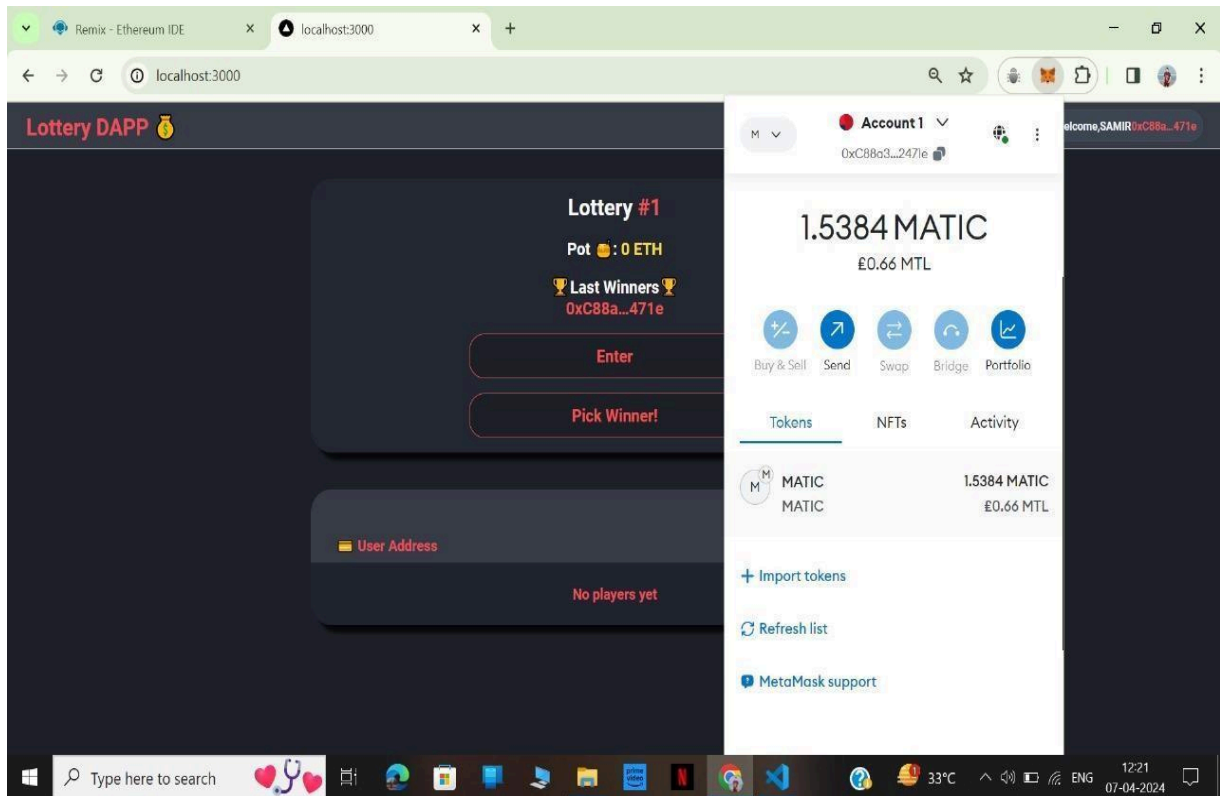


Fig 6.5: Winner got the Amount

CHAPTER 7

CONCLUSIO

N

In conclusion, the implementation of this lottery dapp using Solidity, Next.js 13, and thirdweb has provided a secure and transparent platform for conducting lotteries. The use of smart contracts and blockchain technology ensures that the entire process is decentralized and free from any fraudulent activities. The integration of Metamask wallet and Ethers.js for transactions adds another layer of security to the system. The use of Tailwind CSS has helped in achieving an aesthetically pleasing and user- friendly design.

Overall, this lottery dapp can serve as an excellent platform for conducting fair and transparent lotteries, with the added benefits of security and ease of use provided by blockchain technology. With further development and optimization, this dapp has the potential to become a popular choice for conducting online lotteries.

REFERENCES

- [1] Thomas Barker and Marjie Britz. 2000. Jokers wild: Legalized gambling in the twenty-first century. Greenwood Publishing Group.
- [2] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. 2013. Keccak. (2013).
- [3] S. Bragagnolo, H. Rocha, M. Denker, and S. Ducasse. 2018. SmartInspect: solidity smart contract inspector. In 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE). 9–18. <https://doi.org/10.1109/IWBOSE.2018.8327566>
- [4] Y. Chen, S. Chen, and I. Lin. 2018. Blockchain based smart contract for bidding system. In 2018 IEEE International Conference on Applied System Invention (ICASI). 208–211. <https://doi.org/10.1109/ICASI.2018.8394569>
- [5] K. Christidis and M. Devetsikiotis. 2016. Blockchains and Smart Contracts for the Internet of Things. IEEE Access 4 (2016), 2292–2303. <https://doi.org/10.1109/ACCESS.2016.2566339>
- [6] Itai Dinur, Orr Dunkelman, and Adi Shamir. 2012. New Attacks on Keccak-224 and Keccak-256. In International Conference on Fast Software Encryption.
- [7] M. Mettler. 2016. Blockchain technology in healthcare: The revolution starts here. In 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom). 1–3. <https://doi.org/10.1109/HealthCom.2016.7749510>
- [8] B. K. Mohanta, S. S. Panda, and D. Jena. 2018. An Overview of Smart Contract and Use Cases in Blockchain Technology. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT). 1–4. <https://doi.org/10.1109/ICCCNT.2018.8494045>
- [9] RANDAO. 2018. RANDAO White Book on Provable Fair Random Number Generation.
- [10] S. Rouhani and R. Deters. 2017. Performance analysis of ethereum transactions in private blockchain. In 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS). 70–74. <https://doi.org/10.1109/ICSESS.2017.8342866>
- [11] Emin Gün Sirer. 2017. How Not To Run A Blockchain Lottery.

<http://hackingdistributed.com/2017/12/24/how-not-to-run-a-blockchain-lottery>

APPENDIX

A. SOURCE CODE

```
// SPDX-License-Identifier: MIT
```

```
pragma solidity ^0.8.15;
```

```
contract Lottery {
```

```
    address public owner;
```

```
    address payable[] public players;
```

```
    address[] public winners;
```

```
    uint public lotteryId;
```

```
    constructor() {
```

```
        owner = msg.sender;
```

```
        lotteryId = 0;
```

```
    }
```

```
    function getWinners() public view returns (address[] memory){
```

```
        return winners;
```

```
    }
```

```
    function getBalance() public view returns (uint) {
```

```
        return address(this).balance;
```

```
    }
```

```
    function getPlayers() public view returns (address payable[] memory) {
```

```
        return players;
```

```
    }
```

```
    function enter() public payable {
```

```
        require(msg.value >= .01 ether);
```

```

    // address of player entering lottery
    players.push(payable(msg.sender));
}

```

```

function getRandomNumber() public view returns (uint) {
    return uint(keccak256(abi.encodePacked(owner, block.timestamp)));
}

```

```

function getLotteryId() public view returns(uint) {
    return lotteryId;
}

```

```

function pickWinner() public onlyOwner {
    uint randomIndex = getRandomNumber() % players.length;
    players[randomIndex].transfer(address(this).balance);
    winners.push(players[randomIndex]);
    lotteryId++;
}

```

```

    // Clear the players array. ['player1', 'player2'] 👉 []
    players = new address payable[](0);
}

```

```

modifier onlyOwner() {
    require(msg.sender == owner);
    _;
}
}

```

```
import style from '../styles/PotCard.module.css'
```

```
import truncateEthAddress from
```

```
'truncate-eth-address' import { useAppContext } from
```

```
'../context/context' const PotCard = () => {
```

```
const { lotteryId, lastWinner, lotteryPot, enterLottery, pickWinner } =
```

```
useAppContext()
```

```
return (  
  <div className={style.wrapper}>  
    <div className={style.title}>  
      Lottery{' '}  
      <span className={style.textAccent}>#{lotteryId ? lotteryId : '1'}</span>  
    </div>  
    <div className={style.pot}>  
      Pot 🐼: <span className={style.goldAccent}>{lotteryPot} ETH</span>  
    </div>  
    <div className={style.recentWinnerTitle}>🏆 Last Winners 🏆</div>  
    {!!lastWinner.length ? (  
      <div className={style.winner}>No winner yet</div>  
    ) : (  
      lastWinner.length > 0 && (  
        <div className={style.winner}>  
          {truncateEthAddress(lastWinner[lastWinner.length - 1])}  
        </div>  
      )  
    )}  
    <div className={style.btn} onClick={enterLottery}>  
      Enter  
    </div>  
    <div className={style.btn} onClick={pickWinner}>  
      Pick Winner!  
    </div>  
  </div>  
)  
}  
export default PotCard
```

```
import { createContext, useState, useEffect, useContext } from 'react'
```

```

import Web3 from 'web3'
import createLotteryContract from '../utils/lottery'
export const appContext = createContext()

export const AppProvider = ({ children }) => {
  const [web3, setWeb3] = useState()
  const [address, setAddress] = useState("")
  const [lotteryContract, setLotteryContract] = useState()
  const [lotteryPot, setLotteryPot] = useState()
  const [lotteryPlayers, setPlayers] = useState([])
  const [lastWinner, setLastWinner] = useState([])
  const [lotteryId, setLotteryId] = useState()
  const [etherscanUrl, setEtherscanUrl] = useState()

  useEffect(() => {
    updateLottery()
  }, [lotteryContract])

  const updateLottery = async () => {
    if (lotteryContract) {
      try {
        const pot = await lotteryContract.methods.getBalance().call()

        setLotteryPot(web3.utils.fromWei(pot, 'ether'))

        setPlayers(await lotteryContract.methods.getPlayers().call())

        setLotteryId(await lotteryContract.methods.lotteryId().call())

        setLastWinner(await lotteryContract.methods.getWinners().call())
        console.log([...lastWinner], 'Last Winners')
      } catch (error) {
        console.log(error, 'updateLottery')
      }
    }
  }
}

```

```
}  
}
```

```
const enterLottery = async () => {  
  try {  
    console.log('entering lottery')  
    await lotteryContract.methods.enter().send({  
      from: address,  
      // 0.015 ETH in Wei  
      value: '150000000000000000',  
      // 0.0003 ETH in Gwei  
      gas: 300000,  
      gasPrice: null,  
    })  
    updateLottery()  
  } catch (err) {  
    console.log(err,  
      'enter')  
  }  
}
```

```
const pickWinner = async () => {  
  try {  
    let tx = await lotteryContract.methods.pickWinner().send({  
      from: address,  
      gas: 300000,  
      gasPrice: null,  
    })  
  
    console.log(tx)  
    setEtherscanUrl('https://ropsten.etherscan.io/tx/' + tx.transactionHash)  
    updateLottery()  
  } catch (err) {
```

```
console.log(err, 'pick Winner')
```

```
}
```

```
const connectWallet = async () => {  
  /* check if MetaMask is installed */  
  if (  
    typeof window !== 'undefined' &&  
    typeof window.ethereum !==  
    'undefined'  
  ) {  
    try {  
      /* request wallet connection */  
      await window.ethereum.request({ method: 'eth_requestAccounts' })  
      /* create web3 instance & set to state */  
      const web3 = new Web3(window.ethereum)  
      /* set web3 instance in React state */  
      setWeb3(web3)  
      /* get list of accounts */  
      const accounts = await web3.eth.getAccounts()  
      /* set account 1 to React state */  
      setAddress(accounts[0])  
      setLotteryContract(createLotteryContract(web3))  
      window.ethereum.on('accountsChanged', async () => {  
        const accounts = await web3.eth.getAccounts()  
  
        /* set account 1 to React state */  
        setAddress(accounts[0])  
      })  
    } catch (err) {  
      console.log(err, 'connect Wallet')  
    }  
  } else {  
    /* MetaMask is not installed */  
    console.log('Please install MetaMask')
```

}

```
return (  
  <appContext.Provider  
    value={{  
      address,  
      connectWallet,  
      lotteryPot,  
      lotteryPlayers,  
      enterLottery,  
      pickWinner,  
      lotteryId,  
      lastWinner,  
      etherscanUrl,  
    }}  
  >  
    {children}  
  </appContext.Provider>  
)  
}  
  
export const useAppContext = () => {  
  return useContext(appContext)  
}
```

B. SCREENSHORTS

Ethereum Mainnet ▾





Welcome back!

The decentralized web awaits

Password

Unlock

[Forgot password?](#)

Need help? Contact [MetaMask support](#)

Fig B.1 Metamask Login

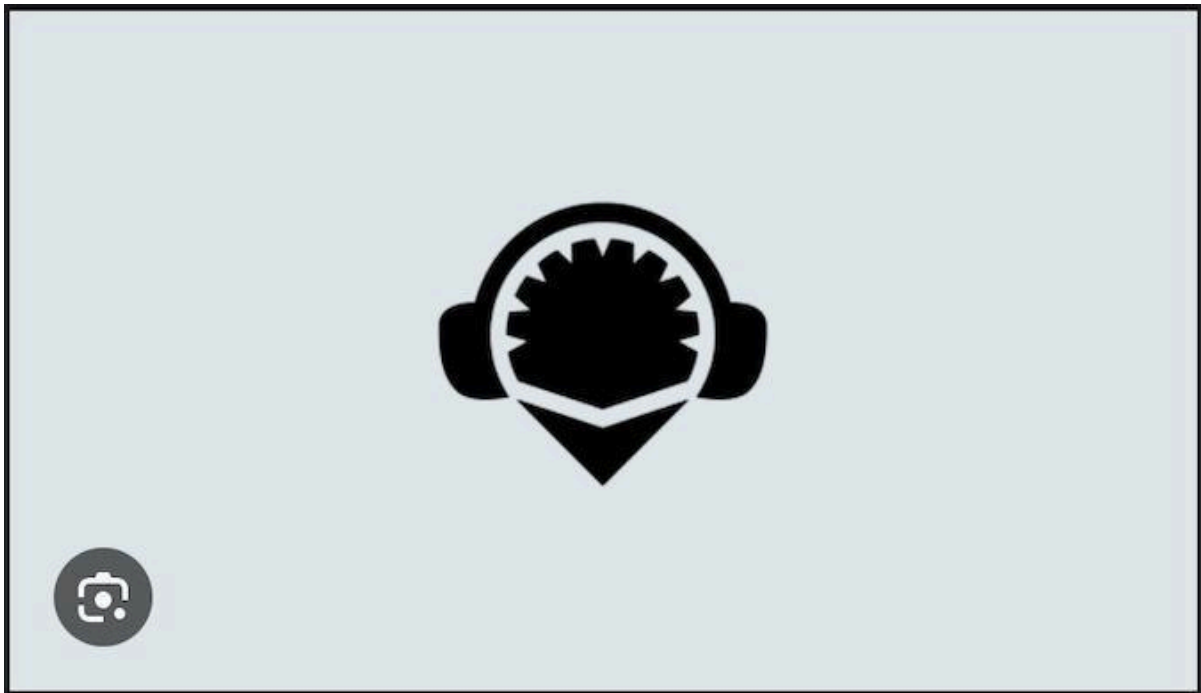


Fig B.2: Remix IDE



Fig B.3: NextJS

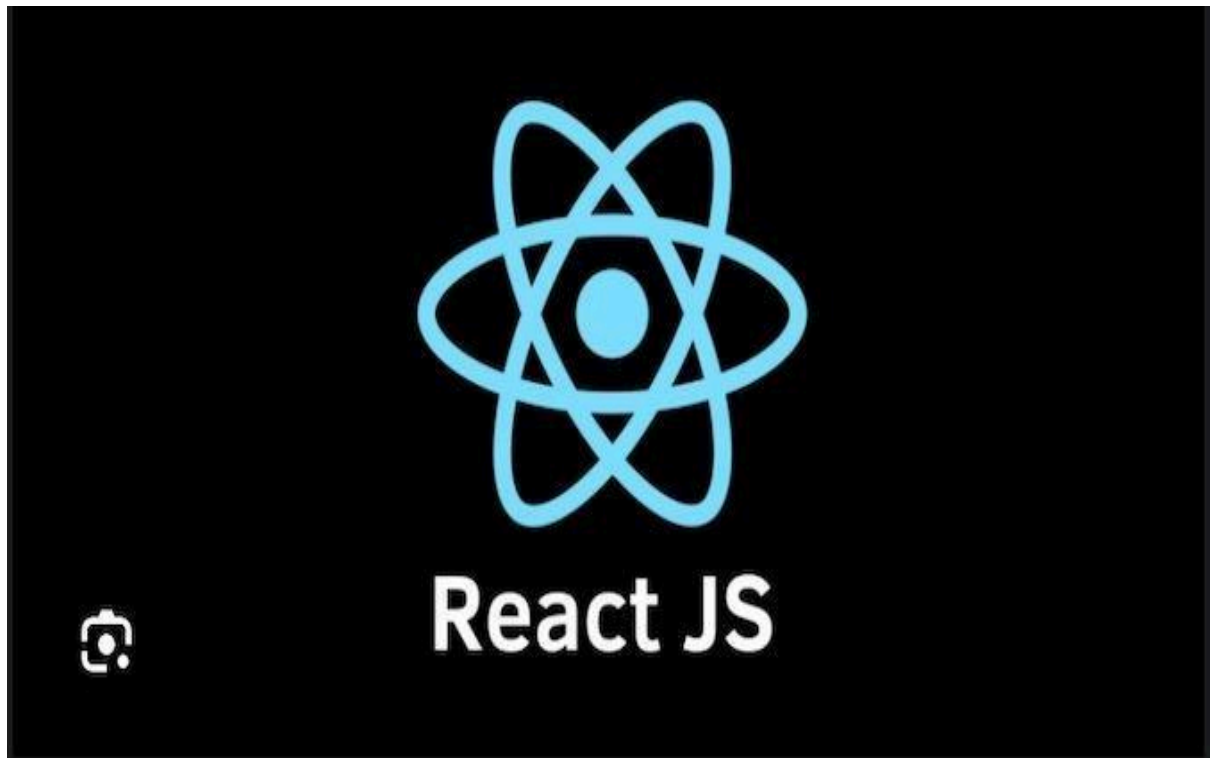


Fig B.4: ReactJS



Fig B.5: TailWindCSS

C. RESEARCH PAPER

DeLottery : Cryptish Lottery App System Based on Blockchain Technology

SONU KUMAR SAW

Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology,
Chennai, India
sonu@sonu.kumar@gmail.com

SAMIIR KUMAR YADAV

Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology
Chennai, India
samyadav8851@gmail.com

Ms. YOGITHA R M.E.,(Ph.D.,)

Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology
Chennai, India
yogitha.ravi1915@gmail.com

ABSTRACT - In Our lottery dapp is designed to allow users to purchase tickets using cryptocurrency and participate in a transparent and fair lottery. This dapp, operating on the Ethereum blockchain, leverages the power of smart contracts to uphold the utmost security and transparency throughout the entire lottery process. The front-end of the application is built using Next.js 13, which provides server-side rendering and automatic code splitting for improved performance and scalability. TailwindCSS is used for styling the application, which provides a range of pre-defined CSS classes to quickly and easily style the interface. The back-end of the application is built using Solidity, which is used to create a smart contract that manages the lottery process. The smart contract includes a function for buying tickets, which accepts cryptocurrency (MATIC) as payment and adds the user's address to the list of participants. Another function is used to randomly select a winner from the list of participants and distribute the winnings to the winner's address. This innovative application utilizes Remix IDE, an intuitive and powerful platform, in tandem with the robust Visual Studio code editor. Featuring cutting-edge programming languages like Javascript and solidity, it harnesses the full potential of the Ethereum blockchain network to effectively store and execute smart contract transactions.

I. INTRODUCTION

Decentralized applications (dapps) are becoming increasingly popular as they leverage blockchain technology to create platforms that are secure, transparent and decentralized. One such application is a lottery dapp, which aims to provide a transparent and secure platform for users to participate in lotteries. Unlike traditional lotteries that rely on a centralized authority to manage and conduct the lottery, lottery dapps utilize smart contracts to ensure fairness and transparency in the selection of lottery winners. Our project is a lottery dapp that is built on the Polygon which is a layer 2 Ethereum blockchain using the Solidity programming language. The dapp allows users to purchase lottery tickets using cryptocurrency and automatically selects a winner based on a random algorithm. The winner is selected in a transparent and fair manner, ensuring that the lottery is free from any manipulation or fraud. Our project aims to demonstrate the potential of blockchain technology in creating a secure and fair platform.

Our groundbreaking platform not only eliminates intermediaries like banks and financial institutions, but also cuts transaction costs and boosts overall efficiency. In this report, we offer an in-depth overview of our lottery dapp's design and implementation. We also evaluate the functionality of the dapp and discuss any issues or challenges encountered during development. The report concludes with a discussion of the potential applications of lottery dapps and their impact on the blockchain industry.

II. LITERATURE REVIEW

[1] This paper has handed me perceptivity about what guarantees does the integration of blockchain and power keep for electricity future. further, more i have found out about how structure plays an vital element in efficient price and we've got set this as our design goal.

[2] This paper stresses quantity decentralization and pseudonymity of the blockchain and has counseled approximately realistic functional and trouble model must be approached as their component is to sort exaggerate the miscalculations and faults inside the design in this case our layout on Lottery using blockchain.

[3] An assault referred to as document denial of chain which employs in a fraudulent way that the blockchain armature and it is settlement mechanisms to be able to control the stored comfy facts by rejecting licit changes or introducing unilateral mechanisms. And we are fastening our interest into enjoin this fault in our layout.

[4] This paper is possibly the primary paper to explain the transactional expenses of implementing blockchain technologies for incipency backing and, putting forward the limitations of blockchain because it does now not use introductory accept ed protocols that made the net extensively reachable like TCP -IP.

[5A] This paper has proven benefits and issues unique Coin presenting[ICO] and original alternate imparting[IEO] via a comparison with fairness conducting the lotteries and we've got allowed about using IOC&IEO as implicit pillars of our layout. Recently, a revolutionary decentralized app was developed on the Ethereum blockchain to tackle the obstacles and simplify the process of making Zakat donations. Being based on the decentralized Ethereum platform, it operates in an asynchronous environment - the internet. To initiate a donation through the dapp, a message is sent using the JavaScript Console and Geth, through the web3.eth.send Transaction command. Due to its asynchronous nature, this process results in improved efficiency and increased success rates.

[5B] The paper discusses about the anomalies incurring all that through offline transaction in blockchain and the way they're reused after connecting to net and testing device of test nets to pretend the EVM and Ethereum community, and we are the usage of outturn comparison of coetaneous, asynchronous and offline sale performed to load and strain testing.

[6] This paper has exceeded me with perceptivity connection between technologies and Financing. There are colourful orders of blockchain grounded on differing types. Blockchains may be permissionless, where participation in the network settlement manner is open to any existent. For policymakers, this quandary is refocused as one it truly is at threat of failing to foster invention thru development, we will try to avoid this dilemma of chance failing.

[7] because blockchain supports a peer-to-peer structure, enables translucency, provenance, and invariability. Peer-to-peer shape is used for the storehouse of facts in every knot. they've a extra allotted records store house which facilitates to store storehouse area. This gets rid of the difficulty of having to host data in massive waiters in a single role. nonetheless, certainly with the 3 algorithms with the loftiest sequestration and security measures, it is continually visible that vicious bumps will always be in truth inside the block chain network. we are able to follow this version the use of smart contracts which solves the authentication trouble. This model provides the sequestration of the specification of a clever settlement, the sequestration of the prosecution of a clever settlement, and the sequestration of the state of a smart agreement.

[8] Decentralization, implementation, and adoption Smart contracts may be expressed by verified information Get out without being in the middle of dramatic events The involvement of a trusted authority figure or important Garson. clever contracts are predicted to revise numerous conventional diligence, similar as financial, operation, IoT, and so forth. nonetheless, there are numerous boundaries additionally which encompass unrecoverable insect s, overall performance issues, lack of depended on data Feeds, lack of norms and rules.

[9] The possibility of operation Blockchain technology in colorful fields remains under take a look at, the decision for investor safety and protection in Lottery contracts may be spoke back by the preface of blockchain technology, which functions on a believe-free gadget where individualities have little to do to make it work. it's clean, comfy, and handy means for the exchange of data and the switch of price range. The era is fluently programmable. however there may be a lack of mindfulness, there is loads of discussion approximately blockchain, but humans don't know the genuine cost of blockchain and the way they could practice it in one-of-a-kind conditions.

[10] the principle advantage of these structures is, human beings can make contributions plutocrat without any trust factor. this is the place in which conventional crowdfunding platforms lacks. every other trouble of the being technique is the centralised waiters, that are difficulty to vicious exertion.

[11] The layout is an internet operation which is basically an improvement of the being transparency in lottery system-

In ordinary Lottery as there's no safety assured for participate quantum. So, clever settlement of Ethereum platform that is an operation of blockchain can be used in order to interrupt this trouble. Blockchain is a decentralized allotted tally machine that accesses, verifies, and transmits community data via distributed bumps.

[12] From this paper what i have found out is we are using Test net and Meta masks. The experience used in this touch is reliability which has surpassed outcomes in which it's been examined on a Ganache unique community which enables for the speed of transactions. The perpetration of this concept contains loopholes which endure correction for the safety of the fund, we are able to try and discover the end result for those loopholes.

[13] smart contracts do no longer require brokers or other intermediaries to validate the agreement, so that they eliminate the chance of manipulation by using 1/3 events, ensuing in fee savings and reduced fraud. clever contracts are assigned, it's miles almost impossible to trade previously assigned ones, and loopholes are feasible. A blockchain -primarily based clever settlement requires a excessive aid fee. smart contract use cryptocurrency, which now not all governments have legalized.

[14] The knowledge I have ate up in this work is that blockchain transactions and clever contracts may be used in lots of areas, starting from coverage claims to the economic quarter. For safety purposes, we may additionally upload patterns to explain positive attacks to assist us apprehend what wishes to be designed to assist the ones assaults.

[15] The use of blockchain technology, smart contracts, and RANDAO [10] is integral to the design of DeLottery. Our system is composed of three major phases: the Initial Phase, Interaction Phase, and Ending Phase, each with two detailed steps. These steps are easily executable using Solidity and guarantee efficiency, with a time complexity no greater than $O(n)$. This unique feature makes DeLottery suitable for a wide variety of applications, particularly those involving large amounts of data.

III. BLOCKCHAIN

Peer-to-peer networking and secure data storage are made possible by blockchain technology. All accounts on the chain are given a private key and a public key, and the technology is based on public-key cryptography. Each encrypted account maintains its private key, while all other accounts share the public key. Using a hash-based digital signature, the recipient of the digital currency starts a transaction by sending his public key to the sender of the funds. In the blockchain, the public keys are regarded as addresses, and every coin has a corresponding address. It is "enabled without disclosing one's identity" which is where the transaction process' novelty lies. As a result, compared to the initial online and offline trading systems, the security of personal information can be reinforced. The following technologies support the features of blockchain: E-signature: Using the private key of the provider's account, each piece of data requires an E-signature. The HASH256 function uses the public key that is specific to it to generate this private key. The private key cannot be found from the public key, but it is almost always possible to verify

public key, but it is almost always possible to verify whether the two are the same using the $\text{HASH}_{256}(\text{private_key}) = \text{HASH}_{256}(\text{public_key})$. Each other user can check this E-signature to confirm that the new data is legitimate once it is uploaded to the blockchain.

IV. EXISTING SYSTEM

In traditional lotteries, the players select some numbers on a ticket, register it with the lottery organizer, and pay a lump sum for it. However, this approach does not provide any assurance for the players that the lottery organizer does not manipulate the number selection in order to make the most money. This suspicion could be eliminated if the lottery organizer did not know the numbers selected by the players prior to the draw. While it is possible to implement such a system using anonymous signatures, the design should also ensure that lottery tickets cannot be forged after the draw or claimed of a different ticket. This paper will propose and analyse a model that meets all the requirements described above, using several cryptographic primitives.

V. PROPOSED SYSTEM

Our design goal is to keep the lottery program close to priority. As lottery is a process as much as possible, avoid unnecessary steps. Here's how we use Smart to DeLottery Agreements on the blockchain. is a higher phase ordering system shown in Figure 1. The players are connected on the blockchain using their digital devices as nodes on the blockchain. The lottery smart contract is implemented on the same blockchain. All players participate in the lottery. They play a role smart contract implementation in single account. Then, the players and the lottery system communicates. Think of it as enough for the users of the lottery system, there is one user of this contract. An arbitrary user and any user can participate in this contract negotiations.

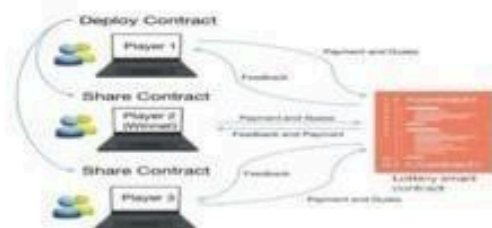


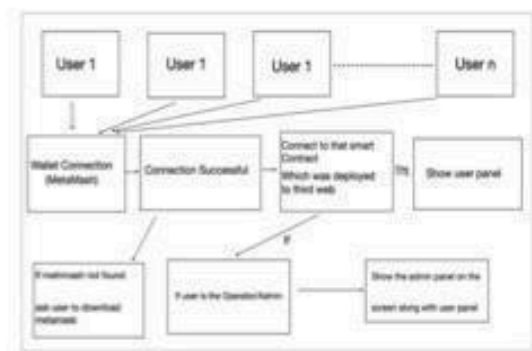
Figure 1: A Lottery System Diagram



Figure 2: State Diagram

At the beginning of the lottery process, the only account used is arbitrary host for DeLottery. We've devised a highly organized finite state machine diagram, shown in Figure 2, to structure our lottery system. DeLottery consists of seven steps, which can be condensed into three distinct phases: the initial phase, the interaction phase, and the ending phase. Each phase is comprised of two steps. Figure 2 depicts different phases in separate rows. Our carefully crafted contract, based on these phases and diagram, should contain multiple variables, as outlined in Table 2. These variables are crucial for storing essential information about users and the lottery system, as well as enabling user interaction. These parameters are configured to support the lottery procedure in its entirety and guarantee the decentralized nature of the system. It includes the specialized variables found in DeLottery in addition to all the variables in Table 1 as is typical of a lottery process.

VI. ARCHITECTURE DIAGRAM



Users can purchase tickets using cryptocurrency and take part in an open and equitable lottery with our lottery dapp. The Ethereum blockchain is the foundation of the dapp, which uses the of smart contracts to guarantee the security and openness of lottery procedure. The application's front end is constructed with Next.js 13, which enhances performance and User openness of the lottery procedure. The application's front end is constructed with Next.js 13, which enhances performance and scalability with serverside rendering and automatic code splitting.

TailwindCSS, which offers a variety of pre-defined CSS classes to quickly and easily style the interface, is used to style the application. Solidity, which is used to create smart contracts that control lottery processes, is used in the construction of the application's back end. A feature of the smart contract allows users to purchase tickets by adding their address to the list of participants and accepting cryptocurrency (MATIC) as payment. The winner is chosen at random from the list of participants using a different function, and their address is given the winnings.

The Dapp's library feature utilizes the powerful web3 (also known as the Ethereum JavaScript API) to enable seamless transaction execution when users interact with the view component, as shown in Figure 4.2. By utilizing Metamask, the library component is able to effectively communicate with Ethereum nodes. Metamask functions as a thin client that offers library components async functionality for calling the Ethereum network through web3-provided APIs. Along with monitoring transaction processing, Metamask also notifies the library component regarding the

the Ethereum blockchain, a transaction must be signed. The web3 library's APIs are used by Metamask to handle this. After the transaction has been signed, Metamask will send it to the Ethereum network, where the smartcontracts are installed, via an RPC call. To update the Library component callback function found in the Library component, MetamaskPolls the Ethereum network to identify transactions position. The library component handles additional network calls to the IPFS for the purpose of storing and retrieving campaign metadata. To communicate with its filesystem, IPFS offers a standard set of APIs. In the future, the library component will only need to make a web call to retrieve the stored data because it will have stored the metadata and received a link to it through web service calls to IPFS. The library component user information gathered from the IPFS and Ethereum network to create a data object, which it then formats so the view component can display it in a browser.

VII. RESULT



Login page.



Connecting Metamask

Users can buy tickets with cryptocurrency and take part in a transparent and equitable lottery with our lottery dapp. To guarantee the security and openness of the lottery process, this dapp is constructed using Ethereum blockchain which uses the smart contracts. The application's front-end is constructed with Next, the server-side rendering and automatic code splitting offered by JavaScript 13 for enhanced scalability and performance.

VIII. CONCLUSION

In conclusion, the implementation of this lottery dapp using Solidity, Next.js 13, and thirdweb has provided a secure and transparent platform for conducting lotteries. This uses the smart contracts and blockchain technology ensures that the entire process is decentralized and free from any fraudulent activities. The integration of Metamask wallet and Ethers.js for transactions adds another layer of security to the system. The use of Tailwind CSS has helped in achieving an aesthetically pleasing and user-friendly design. Overall, this lottery dapp can serve as an excellent platform for conducting fair and transparent lotter-

-ies, with the added benefits of security and ease of use provided by blockchain technology. With further development and optimization, this dapp has the potential to become a popular choice for conducting online lotteries.

IX. ACKNOWLEDGEMENT

I would like to thank the professors of Sathyabama University for which allows me to conduct this study. We would especially like to thank Ms. YOGITHA R. M.E., (Ph.D.) for her support.

X. REFERENCE

- [1] Thomas Barker and Marjie Britz. 2000. *Jokers wild: Legalized gambling in the twenty-first century*. Greenwood publishing Group.
- [2] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. 2013. Keccak. (2013). S. Bragagnolo, H. Rocha, M. Denker, and S. Ducasse. 2018. SmartInspect: solidity smart contract inspector. In 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE). 9-18. <https://doi.org/10.1109/IWBOSE.2018.8327566>
- [4] Y. Chen, S. Chen, and I. Lin. 2018. Blockchain based smart contract for bidding system. In 2018 IEEE International Conference on Applied System Invention (ICASI). 208-211. <https://doi.org/10.1109/ICASI.2018.8394569>
- [5] K. Christidis and M. Devetsikiotis. 2016. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access* 4 (2016), 2292-2303. <https://doi.org/10.1109/ACCESS.2016.2566339>
- [6] Itai Dinur, Orr Dunkelman, and Adi Shamir. 2012. New Attacks on Keccak-224 and Keccak-256. In *International Conference on Fast Software Encryption*.
- [7] Da-Yin Liao and Xuehong Wang. 2017. Design of a Blockchain-Based Lottery System for Smart Cities Applications. In *Collaboration and Internet Computing (CIC)*, 2017 IEEE 3rd International Conference on. IEEE, 275-282.
- [8] M. Mettler. 2016. Blockchain technology in healthcare: The revolution starts here. In 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom). 1-3. <https://doi.org/10.1109/HealthCom.2016.7749510>
- [9] B. K. Mohanta, S. S. Panda, and D. Jena. 2018. An Overview of Smart Contract and Use Cases in Blockchain Technology. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT). 1-4. <https://doi.org/10.1109/ICCCNT.2018.8494045>
- [10] RANDAO. 2018. RANDAO White Book on Provable Fair Random Number Generation.

In 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), 70–74. <https://doi.org/10.1109/ICSESS.2017.8342866>

[12] George Sher. 1980. What makes a lottery fair? *Noûs* (1980), 203–216.

[13] Emin G  jn Sirer. 2017. How Not To Run A Blockchain Lottery. <http://hackingdistributed.com/2017/12/24/how-not-to-run-a-blockchain-lottery/>

[14] F. Tschorsch and B. Scheuermann. 2016. Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Communications Surveys Tutorials* 18, 3 (thirdquarter 2016), 2084–2123. <https://doi.org/10.1109/COMST.2016.2535718>

[15]Henk C. A. van Tilborg and Sushil Jajodia (Eds.). 2011. Proof of Work. Springer US, Boston, MA,984-984.https://doi.org/10.1007/978-1-4419-5906-5_1060[16]E. Yavuz, A. K. Ko  g, U. C.   g abuk, and G.Dalk  sl   g. 2018. Towards secure evoting using ethereum blockchain. In 2018 6th International Symposium on Digital Forensic and Security (ISDFS). 1-7. <https://doi.org/10.1109/ISDFS.2018.8355340>[17]Choi S Park S Smolander K Yli-Huuma J, KoD (Ed.). 2016. .