IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 6, June 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 6, June 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1006167|

Play with Pros Using Tokens Using Blockchain

Shailendra Saurav, Shashwat John, Shyam Jaiswal, Shlok Satyam Dubey, Mr. Suraj Kumar B P

B.E Students, Dept of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bangalore, Karnataka, India

Associate Professor, Dept of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bangalore, Karnataka, India

ABSTRACT: We are making a platform where professional e-sports players can charge tokens to play with customers. These tokens will regulate throughout the application according to demand and supply. Our application will contain different packages and offers to attract customers. For example, some of the packages include, users can ask players to review their match clips, users can ask players to play with them, users can take up subscription for some duration of time to play with players.

KEYWORDS: Blockchain, Deep Learning.

I. INTRODUCTION

Blockchain is a recent technology and has been a buzz since a while. It has numerous use cases and advantages. One of it is being a decentralized system. As we are advancing towards future we have been always seeking easy and efficient way of handling things. We always tend to remove middle man and agent and the charges involving them. This decentralized system has enabled us to achieve a system comprising of peer-to-peer connection together with multiple servers or blocks or nodes to rely on. We will use this technology to enable users or customers to book an appointment with their favourite pro players of any e-sports using tokens and then playing with them to upgrade their skills.

II. PROBLEM STATEMENT

The goal of this project is to develop a play with prosapplication that will people to play with professional players or the players they like or follow.To play with their favorite player they need to give tokens in exchange.

III.METHODOLOGY

A selection of models were used to make this project. Some of these are-

1) React JS

JavaScript library. License. MIT License. Website. reactjs.org. React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies.

2) React Router Dom

React Router DOM enables you to implement dynamic routing in a web app. Unlike the traditional routing architecture in which the routing is handled in a configuration outside of a running app, React Router DOM facilitates component-based routing according to the needs of the app and platform.

3) web 3

Web3 is the name some technologists have given to the idea of a new kind of internet service that is built using decentralized blockchains — the shared ledger systems used by cryptocurrencies like Bitcoin and Ether.

The term has been around for years, but it has come into vogue in the past year or so. Packy McCormick, an investor who helped popularize web3, has defined it as “the internet owned by the builders and users, orchestrated with tokens.”

4) Open Zeppelin

Decentralized applications (daps) are hotter than ever. However, despite decentralized applications’ considerable security advantages, they are still vulnerable to some security breaches. This is something that Open Zeppelin looks

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 6, June 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1006167|

to make up for. This full guide answers the question "what is Open Zeppelin?" and we'll take a closer look at how to integrate features from Open Zeppelin when using Moralis. In addition, we will look at Open Zeppelin smart contracts in relation to the popular ERC-20 token standard.

IMPLEMENTATIONS-

```
async function getBalance() {
  const currentBalance = await contract.methods.balanceOf(accounts[0]).call();
  setBalance(currentBalance);
}

const buyTokens = async (event) => {
  event.preventDefault();
  await contract.methods.buyTokens(tokenMintAmount).send({from: accounts[0], value: web3.utils.toWei(amount, 'ether')});
  await getBalance();
}

// _ticketNumber: uint256
const [ticketNumber, setTicketNumber] = useState(0);

const buyTicket = async (event) => {
  event.preventDefault();

  await contract.methods.buyTicket(ticketNumber).send({from: accounts[0]});
  await getBalance();
}

// _game: String, _play_at: uint256, _duration: uint256, _tokens_required: uint256
const [game, setGame] = useState("");
const [playAt, setPlayAt] = useState(0);
const [duration, setDuration] = useState(0);
const [tokensRequired, setTokensRequired] = useState(0);

const createTicket = async (event) => {
event.preventDefault();

await contract.methods.createTicket(game, playAt, duration, tokensRequired).send({from: accounts[0]});
await viewTickets();

}
```

Fig 1(BALANCE FUNCTION)

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 6, June 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1006167|

```
const [balance, setBalance] = useState(0);

const TOKEN_PRICE = 0.00001;
const amount = (TOKEN_PRICE * tokenMintAmount).toString();

useEffect(() => {
  const init = async () => {
    try {
      // console.log("hi")
      // Get network provider and web3 instance.
      const web3 = await getWeb3();
      // const web3 = new Web3(Web3.givenProvider)
      // if (!(window.ethereum && window.ethereum.isMetaMask)) {
      //   throw Error('Metamask not installed!!!')
      // }

      // await window.ethereum.request({ method: 'eth_requestAccounts' })

      // Use web3 to get the user's accounts.
      const accounts = await web3.eth.getAccounts();

      // Get the contract instance.
      const networkId = await web3.eth.net.getId();
      const deployedNetwork = Contract.networks[networkId];
      const instance = new web3.eth.Contract(
        Contract.abi,
        deployedNetwork && deployedNetwork.address,
      );
      // Set web3, accounts, and contract to the state, and then proceed with an
      // example of interacting with the contract's methods.
      setWeb3(web3);
      setAccounts(accounts);
      setContract(instance);

      // current token balance of the user
      const currentBalance = await instance.methods.balanceOf(accounts[0]).call();
      setBalance(currentBalance);
```

Fig 2(TOKEN FUNCTION)

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 6, June 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1006167|

```
function buyTicket(uint256 _ticketNumber) public returns(bool) {

    require(balanceOf(msg.sender) >= tickets[_ticketNumber].tokens_required, "account does not have
    require(transfer(tickets[_ticketNumber].creator, tickets[_ticketNumber].tokens_required), "canno
    tickets[_ticketNumber].buyer = msg.sender;
    return true;
}

function viewTicket(uint256 _ticketNumber)
            public
            view
            returns(
                address,
                address,
                string memory,
                uint256,
                uint256,
                uint256,
                uint256) {

    return (
        tickets[_ticketNumber].creator,
        tickets[_ticketNumber].buyer,
        tickets[_ticketNumber].game,
        tickets[_ticketNumber].created_at,
        tickets[_ticketNumber].play_at,
        tickets[_ticketNumber].duration,
        tickets[_ticketNumber].tokens_required
            );
}
```

Fig 3(BUYING FUNCTION)

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 6, June 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1006167|

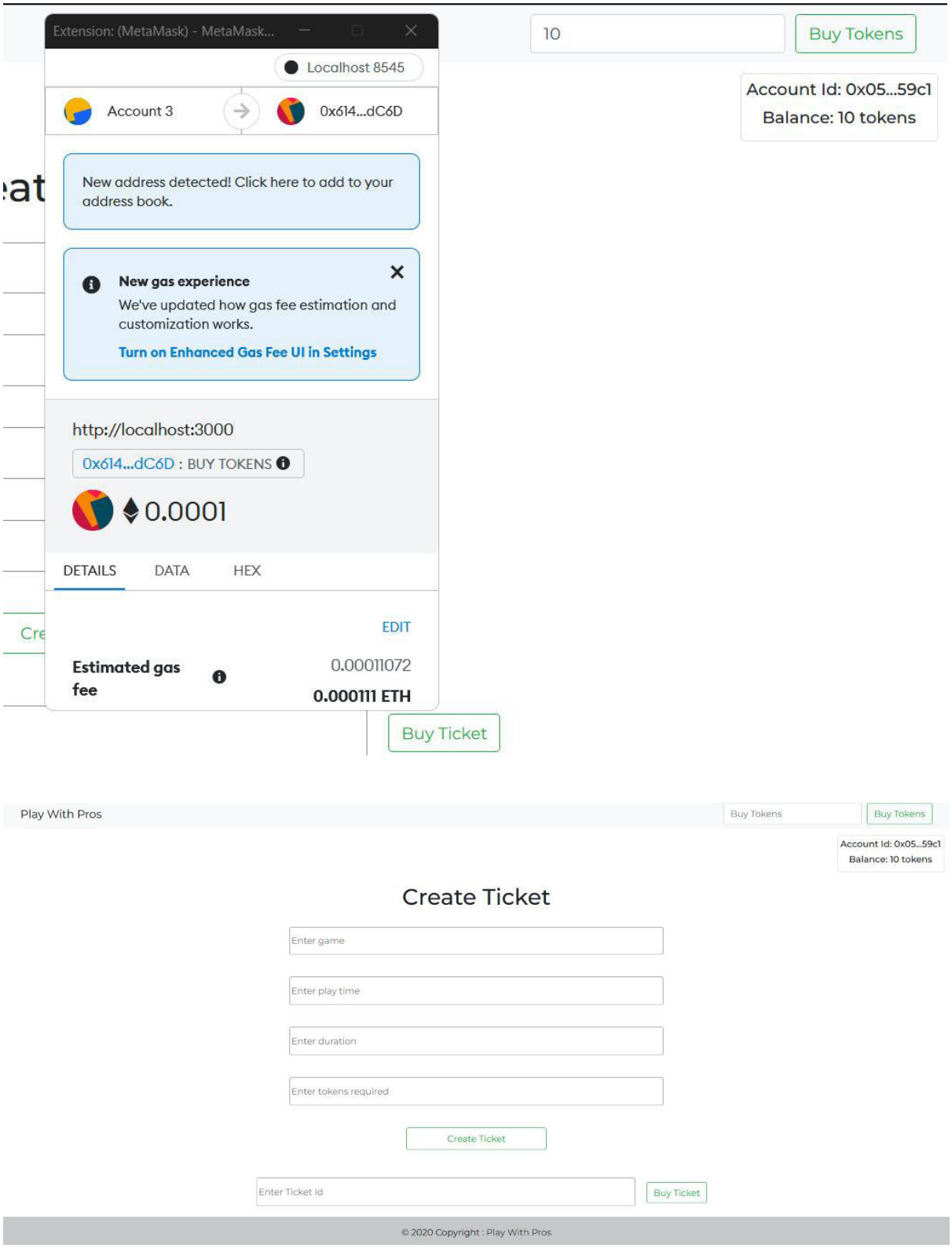

Fig 4(UI)

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 6, June 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1006167|

V. CONCLUSION

As we are advancing towards future we have been always seeking easy and efficient way of handling things. We always tend to remove middle man and agent and the charges involving them. This decentralized system has enabled us to achieve a system comprising of peer-to-peer connection together with multiple servers or blocks or nodes to rely on. We will use this technology to enable users or customers to book an appointment with their favourite pro players of any e-sports using tokens and then playing with them to upgrade their skills

REFERENCES

1. www.reactjs.org
2. www.google.com
3. www.stackoverflow.com
4. www.bootstarp.com
5. docs.soliditylang.org
6. github.com/ethereumbook/ethereumbook
7. Blockchain Basics book by Daniel Drescher

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.165

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details