IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DECENTRALIZED E-COMMERCE MARKET PLACE USING BLOCKCHAIN

[1]Katta Yuktha Sree, [2]Bhogi Monisha, [3]Shaik Shahana Yasmeen chai, [4]Chaitanya Sree Devalla, [5]K. Madhavi

[1]student, [2]student, [3]student, [4]student, [5]student

[1]Gokaraju Rangaraju Institute of Engineering and Technology ,

[2]Gokaraju Rangaraju Institute of Engineering and Technology,

[3]Gokaraju Rangaraju Institute of Engineering and Technology,

[4]Gokaraju Rangaraju Institute of Engineering and technology,

[5]Gokaraju Rangaraju Institute of Engineering and technology

ABSTRACT

The decentralized blockchain e- commerce marketplace project aims to transform traditional online retail by utilizing blockchain technology to eliminate middlemen, boost security, and increase transparency. By leveraging distributed ledger technology, smart contracts, and decentralized storage systems, the platform facilitates secure transactions and maintains data integrity. Its user-friendly interface encourages direct interaction between buyers and sellers, removing the reliance on centralized entities. Smart contracts automate tasks such as order processing, payment handling, and dispute resolution, simplifying transaction procedures. The project targets common issues in conventional e-commerce platforms like high fees, opaque processes, and data vulnerabilities. Through blockchain adoption, it creates a secure and transparent environment, supporting diverse goods and services. It aims to lower entry barriers for small businesses, spur economic growth, and enable global trade. Security measures, including encryption protocols and decentralized storage, safeguard sensitive data, enhance user privacy, and combat fraud and counterfeit products. Success depends on adopting cutting-edge blockchain solutions, forming strategic partnerships, and engaging with the community. Continuous development and research are crucial to adapt to user needs and market changes. Ultimately, the project strives to revolutionize ecommerce by fostering a secure, transparent, and inclusive platform for international trade.

CHAPTER 1

INTRODUCTION

Traditional e-commerce platforms have experienced substantial growth in recent years, becoming prominent players in the global market. Companies like Amazon, Alibaba, and Shein have played a significant role in shaping the e-commerce landscape and driving sales to unprecedented levels [1]. These often suffer from centralization, limited privacy, low settlement times, lack of transparency, and susceptibility to fraud.

In recent years, blockchain technology has transformed various industries, and e-commerce is no exception. A decentralized e-commerce marketplace, powered by blockchain, offers a revolutionary approach to online buying and selling. Blockchain, the technology behind Bitcoin, has evolved beyond currency transactions [2] which acts as a distributed ledger, ensures transparency, security, and decentralization by recording transactions across multiple nodes in a tamper-proof manner. Blockchain operates on a decentralized network of nodes, where each transaction is recorded in a block and cryptographically linked to the previous one, forming a chain. This immutable and transparent ledger ensures that every transaction is verifiable and resistant to tampering, providing a high level of trust in the system. Decentralized marketplaces address issues of traditional e-commerce platforms by enabling direct peer-to-peer transactions without intermediaries. By leveraging blockchain technology, buyers and sellers can engage directly, eliminating the need for third-party intermediaries and reducing transaction costs.

The use of blockchain in ecommerce provides several advantages. Firstly, blockchain ensures transparency and immutability, as every transaction is recorded on a public ledger that is accessible to all participants. This transparency reduces the risk of fraud and counterfeit goods, as buyers can verify the authenticity of products before making a purchase. Secondly, blockchain enhances security by employing cryptographic techniques to secure transactions and prevent unauthorized access. Each transaction is encrypted and linked to the previous one, making it nearly impossible for hackers to alter the data. Furthermore, blockchain enables decentralization, as the ledger is distributed across a network of nodes, rather than being controlled by a single entity. In addition to these advantages, blockchain technology offers innovative features such as smart contracts and decentralized reputation systems. Smart contracts are self-executing contracts with the terms of the agreement directly written into code. They automate the execution of transactions, reducing the need for intermediaries and streamlining the purchasing process. Decentralized reputation systems enable users to build trust based on transparent feedback and ratings, further enhancing the integrity of the marketplace.

Ethereum is established in 2015 and its role as an upgraded cryptocurrency, introduces advanced features and its potential for innovation in both financial and nonfinancial domains. It supports smart contracts and decentralized applications (dApps) and Ether (ETH), its native cryptocurrency, facilitates transactions and smart contract execution. [3,4,5]

Polygon network which is a layer-2 scaling solution for Ethereum network offers faster and cheaper transactions, enhanced security, improved user experience and also facilitates interoperability. It provides scalability through

sidechains, Plasma chains, and the Polygon PoS consensus mechanism, enabling developers to create dApps with Ethereum compatibility.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 SOFTWARE REQUIREMENTS

The system's software components and interfaces are described in detail in the software requirements. These are some software requirements:

1. Operating System : Mac, Linux, Windows
2. Programming Language : JavaScript, Solidity
3. Libraries used : React
4. Blockchain : Polygon Mumbai Testnet

CHAPTER 3

LITERATURE SURVEY

The Researchers introduced a decentralized asset marketplace addressing complexities in NFT [6] transactions by eliminating middlemen and enhancing security. Leveraging React and web3.js, it offers a seamless interface for users to trade NFTs transparently and securely on the Ethereum blockchain. Future plans include integrating with other blockchains and enhancing user experience features.

Hamed Taherdoost et al. [7] examined the potential of blockchain technology to address issues such as payment disputes, fraud, and lack of transparency in conventional e-commerce by conducting a thorough analysis of existing literature on blockchain-based e- commerce, highlighting the benefits and challenges of implementing blockchain in online transactions and also emphasizing the need for further research and development in this field and suggests that blockchain has the potential to improve transaction efficiency, security, and product transparency in e-commerce. They acknowledged that challenges such as scalability and regulatory issues need to be addressed for successful integration of blockchain technology into e-commerce platforms.

Ebru Aydoğan et al. [8] examined the challenges faced by the e-commerce industry, including privacy, security, transparency, and costs, exacerbated by factors like the COVID-19 pandemic and evolving consumer preferences. They did a review and analysis of existing literature on blockchain technology's potential applications in ecommerce. They concluded that blockchain offers significant advantages such as enhanced security, low transaction costs, and rapid speeds, which could revolutionize e-commerce platforms and also acknowledged the need for further investigation into potential risks and technical considerations,

such as scalability and regulatory compliance, to maximize the effectiveness of blockchain-based e-commerce solutions.

Meshari Aljohani et al. [9] are addressing challenges in establishing decentralized data marketplaces, focusing on ensuring data confidentiality, integrity, and trustworthy transactions. They review literature on obstacles and enabling technologies for such marketplaces and propose a framework based on blockchain technology. Smart contracts facilitate secure data exchanges, with a unique approach of using security deposits to enforce honesty. However, potential scalability issues, data accuracy concerns, and legal/regulatory considerations are not thoroughly examined. Further research is needed to assess the framework's scalability, security, and real-world applicability through testing and collaboration with industry partners.

Mohammed Huzaif et al. [10] present their research on decentralized e-commerce using blockchain technology, focusing on the significance of reliability, security, and user privacy in online transactions. They emphasize the necessity of trustworthy sellers and authentic products, particularly given the impact of the Covid-19 pandemic on global retail sales. Exploring blockchain technology and smart contracts, they highlight their potential to enhance security and transparency in e-commerce. They also compare traditional and blockchain-integrated e-commerce systems, highlighting the advantages of the latter, such as improved supply chain management and reduced transaction fees. Ultimately, they conclude that integrating blockchain technology can overcome the limitations of traditional e-commerce, creating a more decentralized, secure, and efficient system.

CHAPTER 4

PROPOSED APPROACH, MODULE DESCRIPTION AND UML DIAGRAM

PROPOSED APPROACH:

Existing centralized marketplaces suffer from issues such as arbitrary decisions affecting merchants, high fees eating into profits, and lack of control over user data. A decentralized marketplace on the Polygon network addresses these by removing intermediaries, automating transactions with smart contracts, and ensuring user data ownership and privacy. This model empowers sellers, reduces costs, and enhances scalability, ultimately creating a more efficient and transparent marketplace ecosystem.

The generalized architecture of a Polygon network is provided in Fig. 1 [11] and an architecture diagram specific to the application being developed is represented in Fig. 2 [12].

A Polygon (Ethereum) network is composed of a set of nodes running an Ethereum client. Each of these

nodes has a copy of the blockchain, which contains a list of all operations performed on the network. This enables nodes to prevent fraudulent activities, such as counterfeiting and duplicating cryptocurrencies, as well as containing an auditable record of all transactions performed on the network. Due to the decentralized nature of the network, direct transactions between users, removing intermediaries for transparency and trust. Decentralized identity and encryption techniques like ECC and SHA-256 where they are provided by the underlying blockchain protocol or associated development framework censure user data privacy and security.

Ethereum provides an interface for consumers through blockchain-enabled extensions and browsers such as MetaMask for Google Chrome and the Mist browser. These interfaces provide a consumer access to their Matic wallets, enabling them to use Polygon to pay for goods.

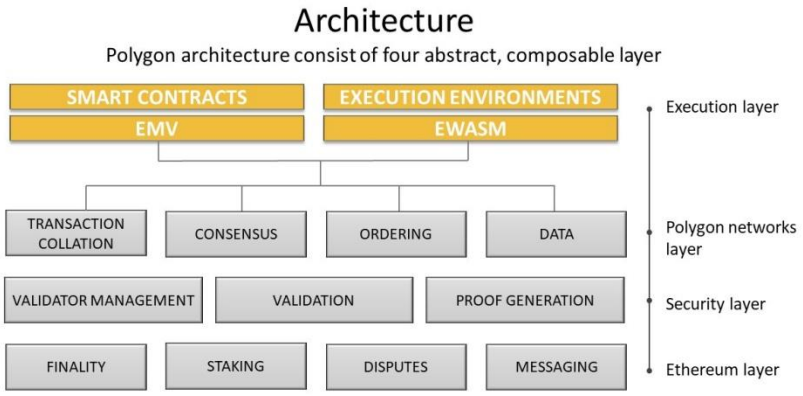

Fig 4.1: Architecture of Polygon network [11]

The above figure is the Polygon (formerly Matic Network) ecosystem operates on multiple layers, each serving a specific purpose within the network. It has the following layers:

a. Ethereum Layer:

Ethereum Mainnet, serving as the foundational blockchain layer for executing smart contracts and storing immutable data. Ethereum's consensus mechanism, transitioning from proof-of-work (PoW) to proof-of-stake (PoS) through Ethereum 2.0 upgrades, provides security and decentralization. Polygon leverages Ethereum as its base layer, benefiting from its robust ecosystem and developer community.

b. Polygon Network Layer:

This layer constitutes the primary layer within the Polygon ecosystem. Operating as a PoS blockchain, this layer facilitates most transactions and smart contract executions, offering fast and low- cost transactions ideal for decentralized applications (dApps) and value transfers. Validators play a pivotal role by staking MATIC tokens as collateral to secure the network and validate transactions, ensuring its security and reliability.

c. Security Layer:

It enhances the security of the Polygon ecosystem by providing checkpoints on Ethereum Mainnet. Validators periodically anchor data from the Polygon Network Layer onto Ethereum Mainnet, safeguarding the integrity and immutability of transactions processed on the Polygon Network Layer. It serves as a vital bridge between the Polygon Network Layer and Ethereum Mainnet, bolstering the security and trust lessness of the Polygon ecosystem.

d. Execution Layer:

Here, smart contracts and decentralized applications (dApps) find their execution environment within the Polygon ecosystem. Smart contracts deployed on the Polygon Network Layer interact within this layer, interacting with the blockchain network and other smart contracts. Developers leverage this layer to deploy their dApps and smart contracts, capitalizing on Polygon's scalability, low transaction fees, and developer-friendly environment. Together, these layers form the Polygon ecosystem, providing scalability, interoperability, and usability enhancements over the Ethereum blockchain, fostering innovation across various decentralized application domains.

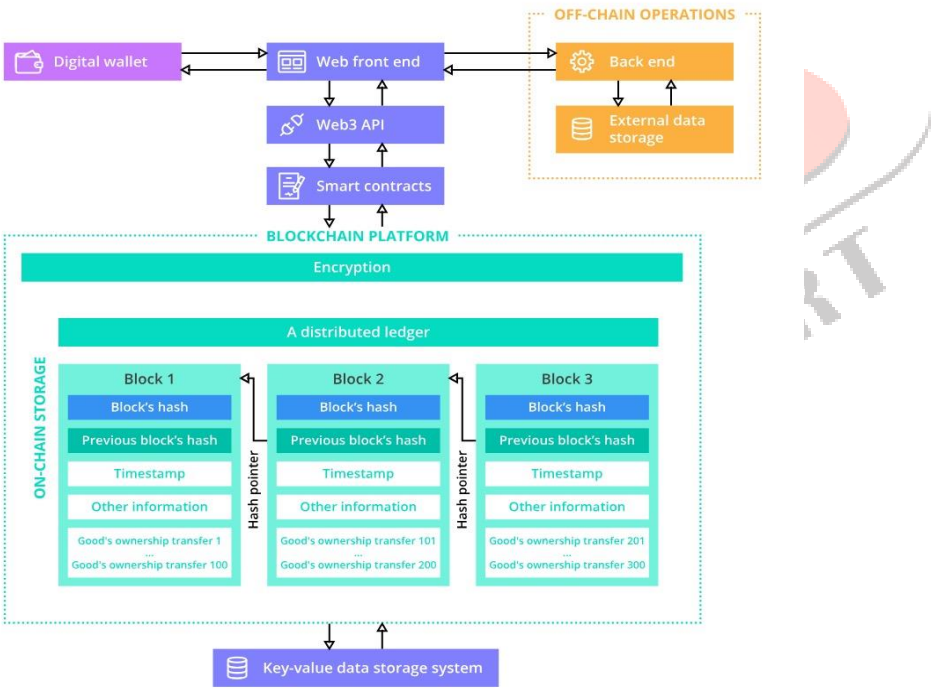

Fig 4.2: Architecture of E-commerce model [12]

The above figure shows the components of the E-commerce model. Each component is explained below.

a. Digital wallet is an app to send and receive crypto tokens used for payment transactions on a decentralized marketplace.

b. The front-end layer is a web application accessible for users. It is created for two user roles – sellers and buyers.

c. The back-end layer processes data that will be stored outside the blockchain (e.g., product listings).

d. Web3 API provides the connection between the web front end and smart contracts.

e. Smart contracts enable verification of marketplace transactions between sellers and buyers.

f. The blockchain encrypts and stores transactional data.

g. Key-value data storage system is used to keep the blockchain metadata.

h. External data storage (often IPFS) is used in distributed blockchain-based systems for user-generated content (product listings).

4.1 MODULES DESCRIPTION

User Authentication and Management Module:

Description: This module handles user registration, authentication, and profile management. Functionality: Users can create accounts, log in securely, update their profiles, and manage their account settings.

Product Listing and Management Module:

Description: This module allows sellers to list products and manage their inventory. Functionality: Sellers can add product details, images, pricing, and availability. They can also edit, delete, or temporarily deactivate listings.

Secure Payment Processing Module:

Description: This module ensures secure payment transactions.

Functionality: It integrates with payment gateways to accept various payment options, including credit cards, fiat money, and crypto currencies. It handles payment processing securely, verifies transactions, and updates order statuses.

User-Friendly Interface Module:

Description: This module focuses on creating an intuitive and user-friendly platform. Functionality: It designs the user interface (UI) for both buyers and sellers, ensuring ease of navigation and a responsive design for various devices**.**

Search and Recommendation Engine Module:

Description: This module enhances the shopping experience by improving product discovery. Functionality: It implements a powerful search engine with filters, categories, tags, and recommendation algorithms to help users find products quickly**.**

Smart Contract and Escrow Module:

Description: This module ensures trust and automates transactions. Functionality: It employs blockchain-based smart contracts to handle escrow, automate payment release upon successful delivery, and manage refunds in case of disputes between buyers and sellers.

Blockchain Integration Module:

Description: This module integrates the blockchain technology into the platform. Functionality: It connects to the blockchain network, records transactions, maintains a transparent and immutable ledger, and ensures the security and integrity of data.

Security and Privacy Module:

Description: This module focuses on enhancing the security and privacy of user data and transactions. Functionality: It implements security measures such as encryption, two-factor authentication, and privacy controls to protect user information and transactions.

Scalability and Performance Module:

Description: This module addresses scalability and system performance.
Functionality: It optimizes the platform's infrastructure, database, and server configurations to ensure smooth performance even with increasing user loads.

4.2 UML DIAGRAMS

The detailed description of a UML diagram for decentralized ecommerce marketplace:

Entities:

- **User:** (Buyer, Seller, Administrator)
- **Product:** (ID, Name, Description, Price, Images, Category)
- **Order:** (ID, Buyer, Seller, Product, Quantity, Price, Status)
- **Transaction:** (ID, Order, Timestamp, Cryptocurrency, Address)
- **Smart Contract:** (Escrow, Dispute Resolution, Rating System)

Relationships:

- **User** <-> Order: A User can place multiple Orders.
- **Order**<-> Product: An Order is for one or more Products.
- **Order**<-> Transaction: Each Order has a corresponding Transaction on the blockchain.
- **Product**<-> Smart Contract: Products interact with Smart Contracts for Escrow and Ratings.
- **User**<-> Smart Contract: Users interact with Smart Contracts for Disputes and Ratings.

Use Case Diagram:

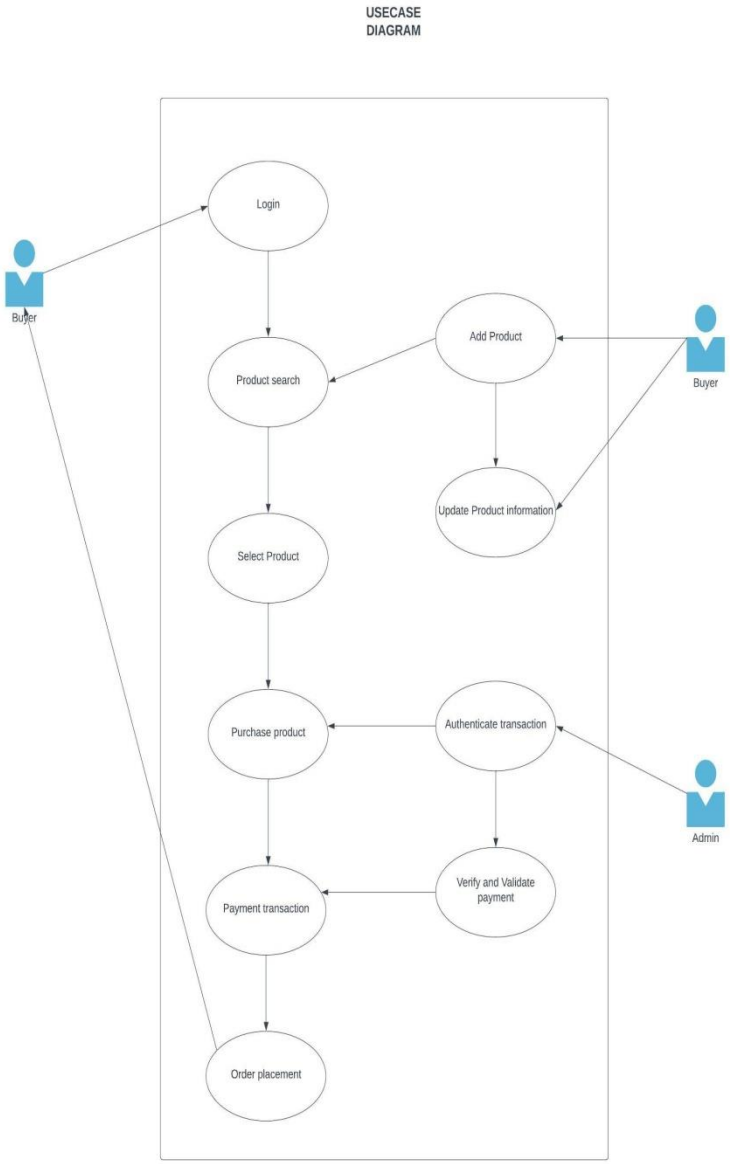

Fig 4.2.1: Use Case Diagram

Activity Diagram:

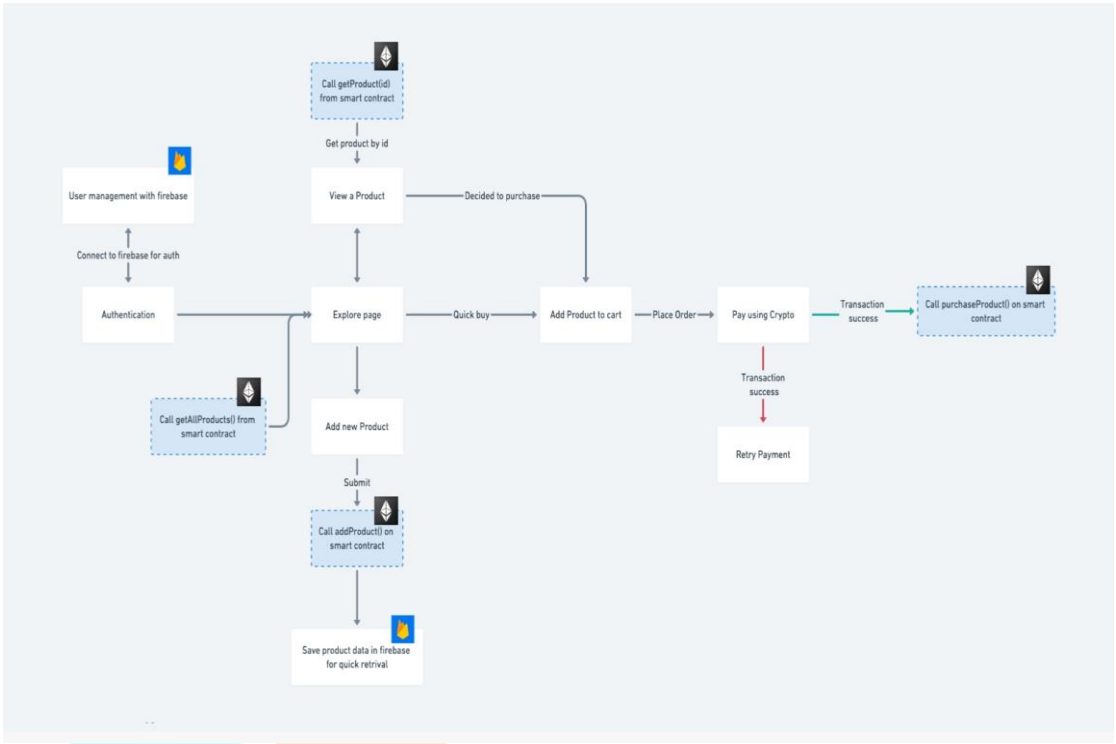

Fig 4.2.2: Activity Diagram

Class Diagram:

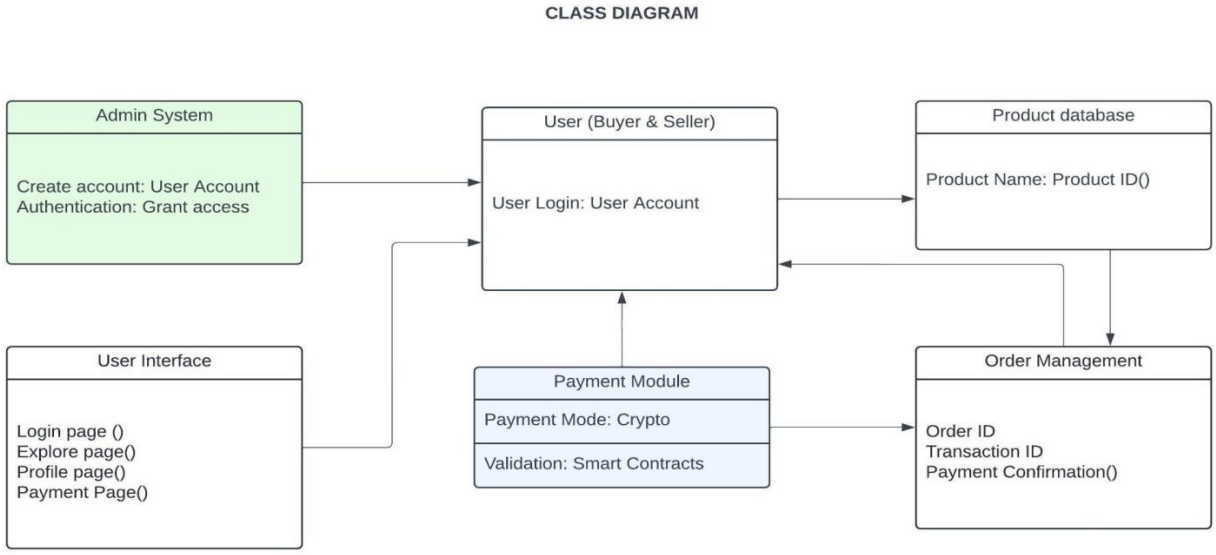

Fig 4.2.3: Class Diagram

CHAPTER 5

IMPLEMENTATION, RESULTS & TEST CASES

IMPLEMENTATION:

Phase 1: Prototype and Core Development:

- **Front-end and Back-end Development:** Build User Interface, API, and Business Logic in JavaScript/React and Node.js.
- **Smart Contract Development:** Implement core contracts for escrow, dispute resolution, and marketplace functionalities using Solidity (Ethereum) or Fabric chaincode.
- **Blockchain Network Setup:** Deploy or join a suitable blockchain network (e.g., Ethereum testnet) for initial trials.
- **Off-chain Storage Integration:** Integrate IPFS or similar decentralized storage for product data and media.

Phase 2: Testing and Refinement:

- **Unit and Integration Testing:** Ensure individual components and their interactions function as intended.
- **Security Audits:** Conduct penetration testing and code reviews to identify and address vulnerabilities.
- **Performance Testing:** Analyze platform scalability and optimize for high user traffic and transaction volume.
- **User Feedback and Iterations:** Gather user feedback through beta testing and refine the platform based on their input.

Phase 3: Deployment and Growth:

- **Mainnet Deployment:** Launch the marketplace on a chosen public blockchain network.
- **Marketing and User Acquisition:** Implement marketing strategies to attract buyers and sellers to the platform.
- **Community Building:** Foster an active user community through forums, tutorials, and incentive programs.
- **Continuous Improvement:** Monitor platform performance, address user feedback, and implement new features and functionalities.

EcommerceContract.sol

C: > Users > saich > AppData > Local > Temp > f9f25d10-56df-4ee6-bef8-b5f496f7ec64_decentralized-eCommerce-main[1].zip.c64 > decentralized-eCommerce-main > contracts > EcommerceContract.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract EcommerceContract {
    address public owner;

    struct Product {
        uint256 id;
        string name;
        string uid;
        string email;
        uint256 price;
        string description;
        address payable account;
        string imgURL;
        uint256 stock;
        uint256 timestamp;
        bool isPurchased;
    }

    uint256 public productCounter;
    mapping(uint256 => Product) public products;

    event ProductAdded(
        uint256 productId
    );

    event ProductPurchased(uint256 productId, address buyer, uint256 amount);
    event ProductDeleted(uint256 productId, address owner);

    modifier onlyOwner() {
        require(msg.sender == owner, "Only the contract owner can perform this action");
        _;
    }

    constructor() {
        owner = msg.sender;
        productCounter = 0;
    }

    function addProduct(
        string memory _name,
        string memory _uid,
        string memory _email,
        uint256 _price,
        string memory _description,
        address payable _account,
        string memory _imgURL
    ) public onlyOwner {
```

Fig 5.1: Smart Contract

App.jsx

C: > Users > saich > AppData > Local > Temp > af14c2cb-ae8d-444b-bf2e-e11287546e4f_decentralized-eCommerce-main[1].zip.e4f > decentralized-eCommerce-main > src > App.jsx

```
import { useEffect, useState } from 'react'
import { Routes, Route } from 'react-router-dom'
import { useGlobalState, setGlobalState, latestPrice } from './store'
import { auth, onAuthStateChanged } from './firebase'
import Product from './views/Product'
import Home from './views/Home'
import SignUp from './views/SignUp'
import SignIn from './views/SignIn'
import AuthGuard from './AuthGuard'
import AddProduct from './views/AddProduct'
import Cart from './views/Cart'
import Chat from './views/Chat'
import ChatList from './views/ChatList'
import { loadWeb3 } from './shared/Freshers'

function App() {
  const [user, setUser] = useState(null)
  const [isLoaded, setIsLoaded] = useState(false)
  const [alert] = useGlobalState('alert')

  useEffect(() => {
    loadWeb3()
    onAuthStateChanged(auth, (user) => {
      if (user) {
        setUser(user)
        setGlobalState('isLoggedIn', true)
      } else {
        setUser(null)
        setGlobalState('isLoggedIn', false)
      }
      setIsLoaded(true)
    })
    latestPrice()
  }, [])

  return (
    <div className='App'>
      {isLoaded ? (
        <>
          {alert.show ? (
            <div className={`text-white px-6 py-2 border-0 rounded relative bg-${alert.color}-500`}>
              <span className='text-xl inline-block mr-5 align-middle'>
                <i className='fas fa-bell' />
              </span>
              <span className='inline-block align-middle mx-4'>
                <b className='capitalize'>Alert!</b> {alert.msg}!
              </span>
              <button onClick={() => setGlobalState('alert', { show: false, msg: '' })} className='absolute bg-transparent text-2xl font-semibold leading-none right-0 top-0 mt-2 mr-6 outline-none focus:outline-none'>
                <span>×</span>
```

Fig 5.2: App Development

```
firebase.js
C: > Users > saich > AppData > Local > Temp > c2ddebe9-7e20-4673-a973-2904c1d9245c_decentralized-eCommerce-main[1].zip.45c > decentralized-eCommerce-main > src > firebase.js
import { initializeApp } from 'firebase/app'
import { setAlert } from './store'
import { getAuth, signInWithEmailAndPassword, createUserWithEmailAndPassword, signOut, onAuthStateChanged } from 'firebase/auth'

import { getFirestore, query, getDocs, updateDoc, collection, collectionGroup, orderBy, deleteDoc, addDoc, doc, setDoc, serverTimestamp } from 'firebase/firestore'
import { contractInstances } from './utils/contracts'
import { BigNumber, ethers } from 'ethers'
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
// console.log(process.env.REACT_APP_FIREBASE_APP_ID, 'api key')
// const firebaseConfig = {
//   apiKey: process.env.REACT_APP_FIREBASE_API_KEY,
//   authDomain: process.env.REACT_APP_FIREBASE_AUTH_DOMAIN,
//   projectId: process.env.REACT_APP_FIREBASE_PROJECT_ID,
//   storageBucket: process.env.REACT_APP_FIREBASE_STORAGE_BUCKET,
//   messagingSenderId: process.env.REACT_APP_FIREBASE_MESSAGING_SENDER_ID,
//   appId: process.env.REACT_APP_FIREBASE_APP_ID,
//   measurementId: process.env.REACT_APP_FIREBASE_MEASUREMENT_ID,
// }

const firebaseConfig = {
  apiKey: "AIzaSyA-s15Q3uY4cAxIHOBpAsUb_GWUIPtRNI4",
  authDomain: "yourburger-95b47.firebaseapp.com",
  databaseURL: "https://yourburger-95b47.firebaseio.com",
  projectId: "yourburger-95b47",
  storageBucket: "yourburger-95b47.appspot.com",
  messagingSenderId: "1055472990706",
  appId: "1:1055472990706:web:6edeed899dd9cd4f68d36b"
}

const app = initializeApp(firebaseConfig)
const auth = getAuth(app)
const db = getFirestore(app)

const logInWithEmailAndPassword = async (email, password) => {
  try {
    return await signInWithEmailAndPassword(auth, email, password).then((res) => res.user)
  } catch (error) {
    setAlert(JSON.stringify(error), 'red')
  }
}

const registerWithEmailAndPassword = async (email, password, fullname, phone, account, address) => {
  try {
    const res = await createUserWithEmailAndPassword(auth, email, password)
    const user = res.user
    const userDocRef = doc(db, 'users', user.email)

    await setDoc(userDocRef, {
      uid: user.uid,
      fullname,
      email,
```

Fig 5.3: Authentication

RESULTS:

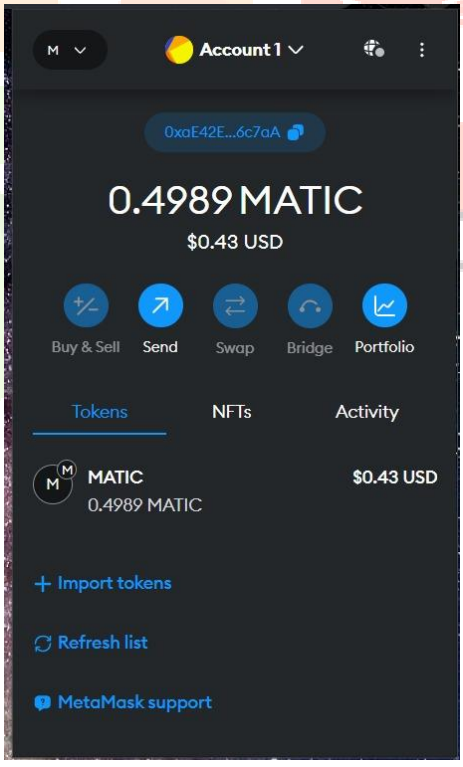

Fig 5.4: Smart Coins

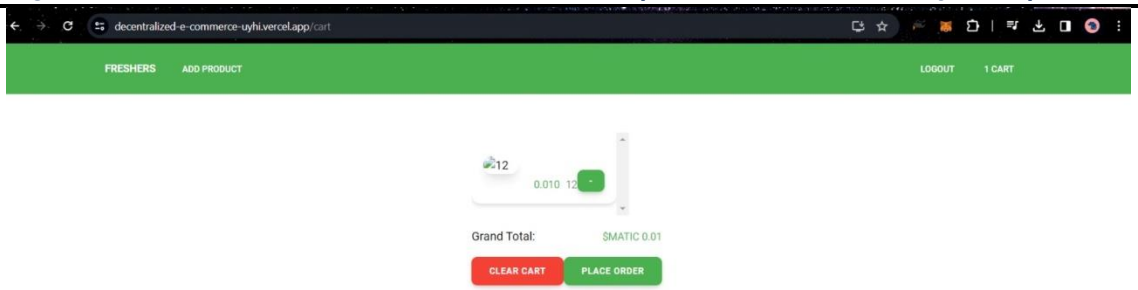

Fig 5.5: Cart

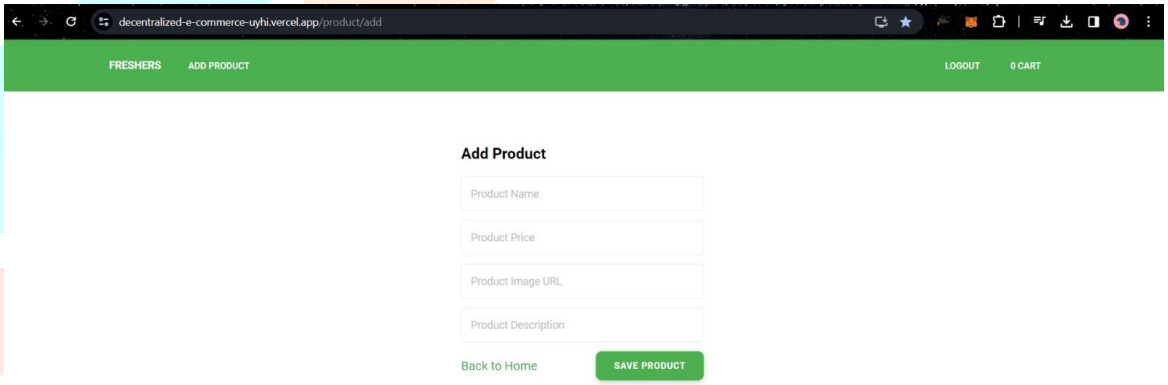

Fig 5.6: Add Product

TEST CASES:

The set of test cases for a decentralized e-commerce marketplace built on blockchain are as follows.

1. User Management:

- **Register/Login:** Successful registration, login, and logout for different user roles (buyer, seller, administrator).
- **Profile Management:** Update user information, profile picture, and settings.
- **User Access Control:** Verify limited access for buyers and sellers, restricted actions for administrators.

2. Product Listing and Management:

- **Product creation:** Verify successful creation of listings with accurate details, images, and pricing.
- **Product search and filtering:** Test search functionality, category filters, and product sorting.
- **Product updates and deletion:** Check if sellers can edit/delete listings within allowed timeframes.

3. Transactions and Smart Contracts:

- **Initiate purchase:** Test buying process, order confirmation, and smart contract execution.
- **Payment processing:** Simulate payment with different cryptocurrencies and verify successful execution.
- **Delivery and dispute resolution:** Test tracking options, dispute initiation, and automated arbitration through smart contracts.

4. Security and Performance:

- **Penetration testing:** Identify and address potential vulnerabilities in the platform's security.
- **Load testing:** Verify platform performance under high user traffic and transaction volume.
- **Blockchain integrity:** Test data persistence and immutability on the blockchain ledger.

5. Decentralization and Scalability:

- **Node functionality:** Test communication between nodes in the decentralized network.
- **Scalability testing:** Check platform performance as user base and transaction volume increase.
- **Resilience against network disruptions**: Simulate node failures and verify fault tolerance of the network.

Module	Test Case	Expected Outcome	Pass/ Fail
Product Listing	1. Seller attempts to list a product with missing information.	Error message indicating missing information.	Pass
	2. Seller successfully lists a product with all required details.	Product listed on platform interface.	Pass
	3. Buyer searches for a product using existing keywords.	Relevant product listings displayed.	Pass
	4. Buyer searches for a non-existent product.	No results found message displayed.	Pass
Purchase Initiation	5. Buyer initiates purchase for a listed product.	Order confirmation screen presented with product	Pass

		details and payment options.	
	6. Buyer attempts to initiate purchase for an out-of-stock product.	Error message indicating product unavailability.	Pass
Payment Authorization	7. Buyer successfully authorizes payment using their wallet.	Transaction confirmation received from wallet and DM platform.	Pass
	8. Buyer declines payment authorization.	Purchase aborted; no funds transferred.	Pass
	9. Buyer wallet connection fails due to network issues.	Error message indicating connection failure.	Pass
Transaction Execution	10. Valid transaction broadcasted to blockchain network.	Transaction successfully added to the blockchain ledger.	Pass
	11. Invalid transaction due to insufficient funds or incorrect signature.	Transaction rejected by the network; no funds transferred.	Pass
	12. Blockchain network congestion delays transaction confirmation.	Pending transaction status displayed until confirmation received.	Pass
Order Confirmation and Fulfillment	13. Seller receives order confirmation notification with buyer details.	Order information displayed in seller account.	Pass
	14. Seller confirms order and marks it for shipment.	Order status updated to "shipped" on DM platform.	Pass
	15. Seller fails to confirm order due to product unavailability.	Order cancelled and buyer notified.	Pass
Delivery Confirmation and Review	16. Buyer receives delivery confirmation notification.	Order status updated to "delivered" on DM platform.	Pass
	17. Buyer submits a positive product review and seller	Review and rating displayed on product listing and seller	Pass

	rating.	account.	
	18. Buyer reports an issue with the product and initiates a dispute resolution process.	Dispute resolution protocol activated; relevant actors notified.	Pass
	19. Buyer fails to leave a review within the specified timeframe.	No action taken; optional review period expires.	Pass

Table 5.1: Test Cases

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

CONCLUSION:

The development of a decentralized blockchain powered commerce platform represents a paradigm change in the field of digital retail. With this initiative, we've effectively combined the benefits of online shopping with the promise of blockchain technology, providing a platform that promotes efficiency, security, and transparency. The distributed ledger technology of the blockchain has enabled us to create a trustless ecosystem, doing away with the requirement for centralized authorities. The security and transparency of transactions, protected by cryptographic methods, are unmatched in traditional e- commerce. In addition, our platform's smart contracts have completely redesigned the enforcement of contracts by automating procedures and guaranteeing term fulfillment without the need for human involvement. This invention puts an end to disagreements, improves dependability, and instills a fresh degree of trust in users. Our platform's user- friendly design and strong backend architecture ensure a smooth experience for both buyers and sellers. The user experience has been our main priority. Blockchain technology has sped up transactions, reduced costs, and offered a decentralized reputation system, all of which have promoted a community-driven mentality. The key to this effort has been teamwork. Our team has put in many hours to get over technological obstacles to guarantee scalability, interoperability, and adherence to changing blockchain standards. As a result, the market is prepared to change and expand inside the ever-evolving blockchain environment. In summary, the blockchain- powered decentralized online marketplace we've created is more than simply a significant accomplishment—it's a demonstration of how technology and innovation can work together. It serves as a lighthouse, shedding light on the way to a time when the characteristics of the digital economy will be efficiency, security, and transparency. This project isn't just the end; rather, it's the beginning of a brand-new era of decentralized trade in which consumers will have the power.

FUTURE SCOPE:

Key areas of future exploration and development include:

1. **Scalability Solutions:** Continued research into scalability solutions for blockchain networks will be crucial to support the growth of decentralized e-commerce marketplaces, ensuring efficient transaction processing and network performance.
2. **Interoperability:** Exploring interoperability between different blockchain platforms and e-commerce ecosystems to facilitate seamless transactions and data exchange across diverse networks.
3. **Regulatory Frameworks:** Addressing regulatory considerations and compliance requirements to foster the integration of decentralized e-commerce marketplaces within existing legal and regulatory frameworks.
4. **User Experience Enhancement:** Further enhancing the user experience of decentralized e-commerce platforms through intuitive interfaces, streamlined processes, and improved accessibility for a broader user base.
5. **Integration of Emerging Technologies:** Leveraging emerging technologies such as AI, IoT, and decentralized finance (DeFi) to augment the capabilities and features of decentralized e-commerce marketplaces.
6. **Community Engagement and Adoption:** Engaging with stakeholders, fostering community participation, and driving adoption initiatives to promote the widespread acceptance and utilization of decentralized e-commerce solutions.

CHAPTER 7

REFERENCES

[1] J. Clement, “Digital retailer from which crossborder digital buyers worldwide made their most recent cross-border digital purchase a September 2023” https://www.statista.com/statistics/878623/ digitalretailers-digital-buyers-cross-border-digitalpurchases/.

[2] S. Nakamoto, “Bitcoin: a peer-to-peer electronic cash system,” March 2009, https://bitcoin.org/bitcoin.pdf.

[3] V. Buterin, “A next generation smart contract & decentralized application platform,” 2015.

[4] W. Gavin, “Ethereum: A secure decentralized generalized transaction ledger byzantium version,” Tech. Rep., 2019.

[5] Jordan Adams, “Revolutionizing Decentralized Applications with Polygon dapps,”, January 2024 https://www.doubloin.com/learn/polygon-dapps.

[6]Shruti Naik, “Decentralized Asset Marketplace,” May 2023, https://doi.org/10.22214/ijraset.2023.51735.

[7] Hamed Taherdoost, Mitra Madanchian, "BlockchainBased E-Commerce: A Review on Applications and Challenges".

[8] Ebru Aydoğan., Muhammed Fatih Aydemir, "BlockchainBased E-Commerce: An Evaluation," 2022, International Journal of Social Inquiry, doi: 10.37093/ijsi.1166643

[9] Meshari Aljohani., Ravindranath, Mukkamala., Stephan, Olariu, "A Framework for a Blockchain-Based Decentralized Data Marketplace," 2023, doi: 10.1007/978-3-031-27041-3_5.

[10] Kavitha K S, "Research on Decentralized E-Commerce Using Blockchain," May 2023, https://www.doi.org/10.56726/IRJMETS38161.

[11] "Matic Network (Polygon) Review: Scaling the Ethereum Blockchain," June 2021, https://hub.forklog.com/obzor-maticnetwork-polygon-masshtabiruya-blokchejn-ethereum/ .

[12]"Architecture of a Blockchain-Based Decentralized Marketplace",2020 https://www.scnsoft.com/blockchain/decentralizedmarketplace#architecture .

[13] Peng Zhang, Michael Walker, Jules White, Douglas C. Schmidt, Vanderbilt University, "Metrics for Assessing Blockchain-based Healthcare Decentralized Apps", IEEE 2018.

[14] G. Bill, A. Freund, J. Bertani, Thomas Cabelguen, B. Towne, D. Kohorn, G. Polzer, P. Karamsetty, M. Bowman, M. Steiner, B. Vavala, T. Willis, J. Katto, P. J. Siemion, C. Nevile, A. Simonet- Boulogne, and H. Croubois, "Enterprise Ethereum Alliance Off- Chain Trusted Compute Specification v1.1," Tech. Rep., 2019.

[15] Alchemy, "Mumbai: Polygon Testnet,", May 2022, https://www.alchemy.com/overviews/mumbai-testnet.

[16] "Cryptocurrency used in polygon network," https://www.coinbase.com/learn/crypto-basics/what-ispolygon .

[17] S. Tamang, ''Decentralized reputation model and trust framework blockchain and smart contracts," 2018, https://www.divaportal.org/smash/get/diva2:1352089/FULLT EXT01.pdf.