PHYSICAL TO DIGITAL SERVICES PROFILING APP

BY

TAN JUN JIE

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) BUSINESS INFORMATION SYSTEMS

Faculty of Information and Communication Technology

(Kampar Campus)

FEBRUARY 2025

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

I want to sincerely thank my supervisor, Dr Aun Yichiet for giving me the chance to work on a blockchain-based system for the first time. Your guidance and support have been incredibly helpful throughout this project.

I am also very grateful to my parents for their financial support, which allowed me to focus fully on my project. Their belief in me has kept me motivated. I also want to thank my friends for their help and encouragement during this time.

Finally, I want to express my deepest thanks to Lee Ern Tong for the constant support and encouragement, especially during the toughest moments. Your support gave me the strength to keep going, and I am truly grateful.

ABSTRACT

In this project, a blockchain-based e-commerce system is proposed to transform the online marketplace by addressing key challenges found in traditional e-commerce platforms. This project's main objective is to eliminate middlemen or intermediaries to streamline transactions, reduce transaction costs, and enhance transparency and trust between buyers and sellers. By utilizing blockchain technology, the system provides decentralized transaction management, user identity verification, and product authenticity. This will provide a secure and efficient platform for e-commerce activities. On the other hand, the user interface is designed to be intuitive and user-friendly to facilitate seamless interaction with the smart contract functionalities. This system will not only redefine the e-commerce experience but also aligns with the sustainable practices by optimizing logistics and minimizing environmental impacts caused by e-commerce activities. This project aims to create an innovative e-commerce platform that is cost-effective, secure and environmentally responsible, which will enhance the user satisfaction and introducing a new era of digital commerce.

Area of Study: Blockchain Technology, Mobile Application Development

Keywords: Blockchain, E-commerce, Flutter, Ethereum, Usability

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

E-Commerce	Electronic Commerce
B2C	Business-to-Consumer
C2C	Consumer-to-Consumer
B2B	Business-to-Business
C2B	Consumer-to-Business
PoW	Proof of Work
PoS	Proof of Stake
IBM	Trademark of International Business Machines Corporation
DIDLT	Decentralized Identifiable Distributed Ledger Technology
BCA	Blockchain Consensus Algorithm
DID	Decentralized Identification
US	United States
TWh	Terawatt-hour
TAM	Technology Acceptance Model
UTAUT	Unified Theory of Acceptance and Use of Technology
TPB	Theory of Planned Behaviour
DOI	Diffusion of Innovation
PEOU	Perceived Ease of Use
PU	Perceived Usefulness
CPU	Central Processing Unit
GB	Gigabyte
RAM	Random-Access Memory
WXGA	Wide Extended Graphics Array
UI	User Interface
IC	Identity Card
enum	enumeration
ETH	Ether
SDK	Software Development Kit
API	Application Programming Interface
Dp	Density-independent Pixels

Px	Pixels
Dpi	Dots Per Inch
ABI	Application Binary Interface
USD	United States Dollar
Wei	Smallest denomination of Ether
VS Code	Visual Studio Code

CHAPTER 1

Introduction

In this chapter, an overview of problem statement and motivation, project scope, project objectives, contribution, and project background were discussed.

1.1 Problem Statement and Motivation

Nowadays, the e-commerce platforms serve as a middleman or an intermediary as they provide a marketplace for sellers and consumers. This has brought challenges such as increased costs, transaction complexities, and possibly trust issues between the platform and its users, affecting the efficiency of online trading. The current e-commerce platforms often impose various fees such as transaction and processing fees which will increase the cost of both buyers and sellers in a transaction. Besides that, the current e-commerce platforms often involve several steps and intermediaries which not only leads to inefficiencies and complexities in the transaction process but also increase the risk of errors. Additionally, the centralized nature of traditional e-commerce platforms can cause trust issues between users and the platform. The platforms are often arbitrary as the buyers and sellers rely on the platform to manage transactions, maintain records and ensure security. This could raise trust issues under some circumstances, for example, when the platform increases the transaction fee, users have to follow the rules as the model is centralized. The middleman model compromises the user satisfaction, adds unnecessary financial burdens, and may prevent users especially the traditional sectors from engaging in the online trading.

This project aimed to address the challenges and eliminate the current intermediaries' model in e-commerce or propose an alternative model. By developing the blockchain-based e-commerce platform, this project aimed to reduce transaction costs, enhance trust between users and platform, and streamline the transaction process by utilizing the blockchain's decentralized nature. This project will help in reducing cost of products and attracting more producer or seller especially those who is in traditional business sector to create a more inclusive digital economy in the world.

1.2 Project Scope

The project scope included the development of a blockchain-based e-commerce platform which will eliminate the middlemen or intermediaries so that it could reduce transaction costs and enhance trust between platform and its users. The feature of the platform includes a decentralized transaction management, user identity verification, and product authenticity check. The user interface will be developed using Flutter to provide a user-friendly experience for users when interacting with the features of platform. The final deliverable will be a functional prototype demonstrating the core functionalities of the system such as user registration, product management, and order management.

1.3 Project Objectives

The objective of this project is to:

1. Propose and implement alternative models to current middleman model in e-commerce, aiming to reduce transaction and processing fees. This will make e-commerce more accessible and cost effective for both buyers and sellers, especially the small and traditional businesses.
2. Design and implement a user-friendly platform that simplifies the transaction process by reducing the steps and intermediaries involved in the process.
3. Implement a decentralized system that provides traceability and transparency of transactions and data management. This project aims to implement blockchain technology to create a secure and transparent platform for users to have more confidence in the platform.

1.4 Contributions

This project could reduce the financial burden by reducing the transaction and processing fees, which makes entering the e-commerce market easier for those small and traditional businesses. This is crucial in creating a more inclusive digital economy with including businesses of all sizes. Besides that, this project could streamline the online transaction process by reducing the intermediaries involved and complexity. This could enhance the efficiency of e-commerce activities, reduce the occurrence of errors, and lead to better overall user experience. The implementation of a decentralized system could significantly improve users' trust in e-commerce platforms. The secured and immutable nature of blockchain technology could make users have confidence in using the platform, promoting a safer and more reliable online marketplace.

1.5 Project Background

In 2023, the global e-commerce market is expected to reach $6.5 trillion in market share and is expected to over $8.1 trillion by 2026 [1]. This indicates that the digitalization of retails is inevitable, and the trend will continue growing. For now, there have been several e-commerce platforms available such as Shopify, Amazon, Taobao, Shopee, Lazada, and the list goes on. Although there have so many platforms existing, more and more new e-commerce platforms still make their appearance such as Temu, which launched in Malaysia in the last quarter of 2023 [2]. However, these e-commerce platforms play a role like a middleman or an intermediary in the business process, which means although they did provide a marketplace for people to trade effortlessly, they will charge tax or processing fee from their users. This intermediary model brings several challenges such as increased transaction costs, reduced transparency of the transaction, and potential trust issues between users and platform.

The intermediary model in e-commerce not only adds financial burdens but also can keep those smaller and traditional businesses away from participating in e-commerce. Moreover, the centralized nature of these existing e-commerce platforms can lead to concerns about data privacy and transparency of transactions on the platforms, which will raise trust issues from the users. An innovation of the current e-commerce model is required to address these issues and move toward a more decentralized and transparent e-commerce model.

In this case, e-commerce could be enhanced by such a new system to provide new ways of making transactions online in a more secure and efficient way. Without the intermediaries in an e-commerce blockchain network, transaction and processing fees will be reduced, and transactions will be smoother and more transparent. Furthermore, the immutable blockchain records of transactions and origin will add authenticity to the product verification, for example, the warranty of the products.

In shorts, this project aimed to propose an innovative solution for the e-commerce industry by eliminating the middlemen through the implementation of blockchain technology. By providing a decentralized, transparent, and secured e-commerce platform, this project seeks to reduce transaction costs, streamline the transaction process, and enhance user trust. This project will create a more efficient and user-friendly e-commerce environment by overcoming the existing challenges faced by traditional e-commerce platforms.

CHAPTER 2

Literature Review

2.1 Previous Studies on Current E-Commerce Models

According to the research [3], the current e-commerce models included Business-to-Consumer (B2C), Consumer-to-Consumer (C2C), Business-to-Business (B2B) and Consumer-to-Business (C2B). the B2C model is dealing with direct sales between business and consumers. The C2C model supports all online trade between consumers, usually through a third party providing an online transaction platform. The B2B model is for business-to-business transactions involve the use of the internet. The C2B model will let consumers offer services or products to businesses, usually through crowdsourcing platforms.

In [4], Santos *et al.* stated that third-party platforms in e-commerce like online auction websites or mobile applications take a commission for themselves without participating directly in the transaction or logistics, which may affect the costs and user experiences. And as most e-commerce platforms involve high investment costs in security and privacy, they are likely to raise transaction costs.

Based on the article [5], one of the fundamental problems of e-commerce is the trust between sellers and buyers since goods and services bought online cannot be immediately verified, and there is still hardly any regulation yet. It is more critical compared to traditional commercial activities. There are also issues in e-commerce security such as integrity of data, authentication, privacy and so on.

2.2 Previous Studies on Blockchain Technology

In an overview of blockchain technology [6], the decentralized nature of blockchain has no central authority and its control is given to a network of nodes. Everything recorded in the blockchain cannot be changed, making every transaction immutable and full of integrity. The article [6] also mentioned that there are two main types of blockchain, which are public and private blockchain. A public blockchain is open to all users. It is highly secure but less efficient. On the other hand, a private blockchain is limited to use with private access and controlled by one organization. It is more efficient but less secure compared to public blockchain. A blockchain system uses programmed smart contracts to execute functions automatically whenever certain events occur. The blockchain network uses consensus algorithms such as Proof of Work (PoW), Proof of Stake (PoS), and so on. Proof of Work is a consensus algorithm used in Bitcoin, where miners are required to solve complex mathematical problems for transactions validation. This algorithm provides security despite it is energy-consuming. Proof of Stake is where the miners validate transactions based on the number of coins they own. It is more energy efficient compared to PoW. The transparency of transactions on blockchain networks will increase trust and decrease intermediaries by the automation of execution.

2.3 Previous Studies on Blockchain in E-Commerce

Research [7] identified several platforms and companies that have been working on blockchains, namely IBM, Maersk, Carrefour, Walmart, and Amazon, where most of them are big players in the commercial sectors. The features of blockchain such as data provenance and traceability has been utilized by companies like Overstock.com and Dubai CommerCity in bringing transparency and efficiency. However, the research [7] mentioned that the blockchain remains a badly researched area in e-commerce, indicating that many challenges to yet being fully identified.

Meanwhile, research [8] also mentioned that blockchain e-commerce is still in the early stage and faces problems relating to data leakage. Based on the research [7] and [8], the blockchain e-commerce model has advantages compared to traditional model, such as less transaction and operational costs by cutting out most intermediaries. Blockchain's decentralized feature provides security and transparency to business transactions where trusted intermediaries are not required, increasing the trust and transparency of e-commerce platforms. It is also mentioned that the security is improved, as blockchain heavily adopts encryption and decentralization which reduces fraud and prohibit unauthorized access. The data stored on blockchain is tamper-resistance.

On the other hand, both research [7] and [8] has raised concerns on the challenges and limitations of blockchain e-commerce system. The research [7] has mentioned that scalability is still one of the biggest challenges, especially in the case of public blockchains, where it might reduce the speed at which transactions are processed due to consensus mechanisms. It might be more problematic when the regulatory and legal considerations are taken into account. For example, different countries with different regulations make it hard to deploy a uniform blockchain solution across borders. In [7], the article also reveals that blockchain is still not understood or commonly accepted within mainstream e-commerce. The struggles with blockchain technology lower the rate of its acceptance into businesses and consumers.

2.4 Previous Studies on Security and Privacy in Blockchain-based E-commerce

According to the research [9], there are technologies in securing e-commerce transactions on blockchain, such as Decentralized Identifiable Distributed Ledger Technology (DIDLT) and the Blockchain Consensus Algorithm (BCA). A distributed ledger is a digital system for recording transactions and data across multiple locations simultaneously. Unlike traditional databases, it doesn’t have a central authority.

These technologies ensure the security of data storage and retrieval through appropriate production of a digital signature, generation of keys, building, and validation of blockchain. The DIDLT-BCA model improved the network security to 98% with the time performance as low as 150 milliseconds and mining at 0.98 seconds.

Each user in the blockchain e-commerce system is uniquely identified with a decentralized identification (DID). A DID is a decentralized identity serving as one’s verified identity securely and digitally. These DIDs are stored on a distributed ledger, which will ensure a tamper-proof identity verification and validation throughout the transaction process.

In [9], it also discussed about the balance between transparency and privacy in blockchain technology. While blockchain provides transparency through its decentralized and immutable nature, cryptographic techniques and DIDs come in to ensure privacy for users. The DIDLT-BCA model provides the necessary transparency in the transactions, making them secure enough to protect user identities from unauthorized access.

On the other hand, it is mentioned in [9] that blockchain e-commerce system could be vulnerable to threats and attacks. Common Attacks like identity theft, payment fraud, and manipulation are mentioned as some of the security threats that might be encountered with the system. Therefore, the DIDLT-BCA model employs various strategies to reduce security risks. For example, Rabin digital signatures is used for block validation to ensure transactions integrity. And BCA is an algorithm that manages incorrect nodes and prevents fraudulent activities.

2.5 Previous Studies on Sustainability and Environmental Impact of E-commerce

Based on the research [10], it mentioned that transportation in logistics significantly contributes to carbon emissions in e-commerce. Although the carbon emissions could be reduced by optimizing the delivery routes, the introduction of same-day and express deliveries, especially with Amazon's Prime services in the lead increased the carbon emissions by delivery vehicles and air transport to accelerate deliveries.

Additionally, most of the e-commerce packaging involves additional layers of boxes, foam, plastic, and bubble wraps to protect items during shipment. The excessive packaging contributes to 30% of the total solid waste produced in the US according to the article [10]. At the same time, most e-commerce operations are designed on a large scale and require intense energy. This mainly caused by maintaining data centres for storing, processing, and retrieving huge volumes of data. According to the International Energy Agency [11], data centres consumed an estimated 240-340 TWh in 2022, accounting for 1 to 1.3% of global electricity demand. The high energy usage of data centres has brought a significant environmental impact.

Besides, the research [12] identified the high usage of energy by blockchain networks, where most of them are based on a consensus algorithm called Proof of Work. For example, it is estimated that Bitcoin alone uses in the range of 30 TWh to 118 TWh annually. It is mentioned in the paper that the transition of Ethereum from the power-consuming PoW to greener solution by shifting into Proof of Stake to address this issue.

Furthermore, the research [12] also mentioned that blockchain can help to improve the logistics and supply chain management. It enhances the traceability and transparency, which makes the supply chains more functional and sustainable. Blockchain also brings full traceability to the products. This will make sure sustainability practices are applied throughout the products' lifecycle.

2.6 Previous Studies on User Adoption and Behavioural Towards Blockchain

In the research [13], several models and theories are applied to assess user's adoption of blockchain technology. This includes the Technology Acceptance Model (TAM), Unified Theory of Acceptance and Use of Technology (UTAUT), Theory of Planned Behaviour (TPB), and Diffusion of Innovation (DOI). These models help understand the factors that influence users' decision to adopt new technologies like blockchain. The research [13] mentioned that Perceived Ease of Use (PEOU) and Perceived Usefulness (PU) are critical factors in the adoption of blockchain technology. For example, in the application of the TAM model, ease of use and usefulness were among the fundamental reasons for the systems to be adopted and used. Meanwhile, the research [14] mentioned that users will believe that blockchain technology is useful if it helps reduce costs and provides reliable information tracking.

According to the research [14], blockchain's features like data privacy security and traceability are highlighted as crucial for building trust. These features ensure data integrity and transparency, which are essential to make user confident in blockchain e-commerce transactions.

Both research [13] and [14] mentioned that education and awareness are key in the mass adoption of blockchain technology. Users can be educated on how blockchain works and how can the blockchain brings benefits to them. The research [14] provided a specific example of giving education with tools such as short videos on how user can understand benefits associated with blockchain.

According to the research [13], the marketing and communication is highly needed to promote blockchain e-commerce platforms. It is important to have an effective marketing and communication on the benefits of blockchain and how can it solve existing real-world problems to attract users.

CHAPTER 3

Proposed Method and Approach

3.1 Design Specifications

3.1.1 Methodologies and General Work Procedures

In this project, the blockchain e-commerce platform was developed using Agile development methodologies. This ensured continuous improvement and adaptation in every stage of the life cycle of the project. This methodology will provide regular feedback and flexibility in the development process to accommodate changes as the development goes on. The major phases of the project include system design, development of smart contracts, integration of the front-end, and deployment of the system.

In system design phase, a detailed system design and the overall architecture of the system was designed. This included the definition of different kinds of roles in the system, interaction points between the different parts of the system such as blockchain backend and front-end interface, the crypto wallet extension used for transactions and so on.

In smart contracts development phase, the smart contracts were developed in Solidity, the language for Ethereum blockchain-based platforms. The contracts were designed to handle important functions such as user registration, product management, and order management. Each smart contract was developed, tested, and iterated to correct and optimize the functions before integrating them into the whole system.

Once the smart contracts were developed, the project should move on to the front-end integration phase. A web page will be developed as the front-end interface prototype, as it is easier and quicker to be developed to have a test on the functionality of the system. The functions should be available on the front-end prototype include user registration, product browsing, product management, and order management.

In the testing and verification phase, unit tests will be individually conducted for each smart contract to verify the functionality and security of the smart contracts. The integration of the smart contracts and front-end will be tested to ensure a user-friendly experience.

In the deployment and evaluation phase, the blockchain e-commerce platform will be deployed and evaluated by selected users. It involved performance evaluations to fulfil several metrics like transaction speed, scalability, and security. Performance under real-world conditions will be observed. User feedback or operational data will be collected to make any improvements to the system.

3.1.2 Tools to Use

Hardware Tools

Description	**Specification**
CPU	x86_64 CPU with at least 4 cores
Memory	Minimum of 8 GB RAM
Display Resolution	Minimum WXGA (1366 x 768 pixels)
Free Disk Space	At least 11 GB of free disk space

Table 3-1-2-1 Table of Hardware Tools

Infrastructure

Description	**Specification**
Internet Connection	High-speed and reliable internet

Table 3-1-2-2 Table of Infrastructure

Software Tools

The detailed data is presented in Appendix Table A-1.

3.2 System Design

This section includes several diagrams which describes the system.

3.2.1 Top-down System Flowchart

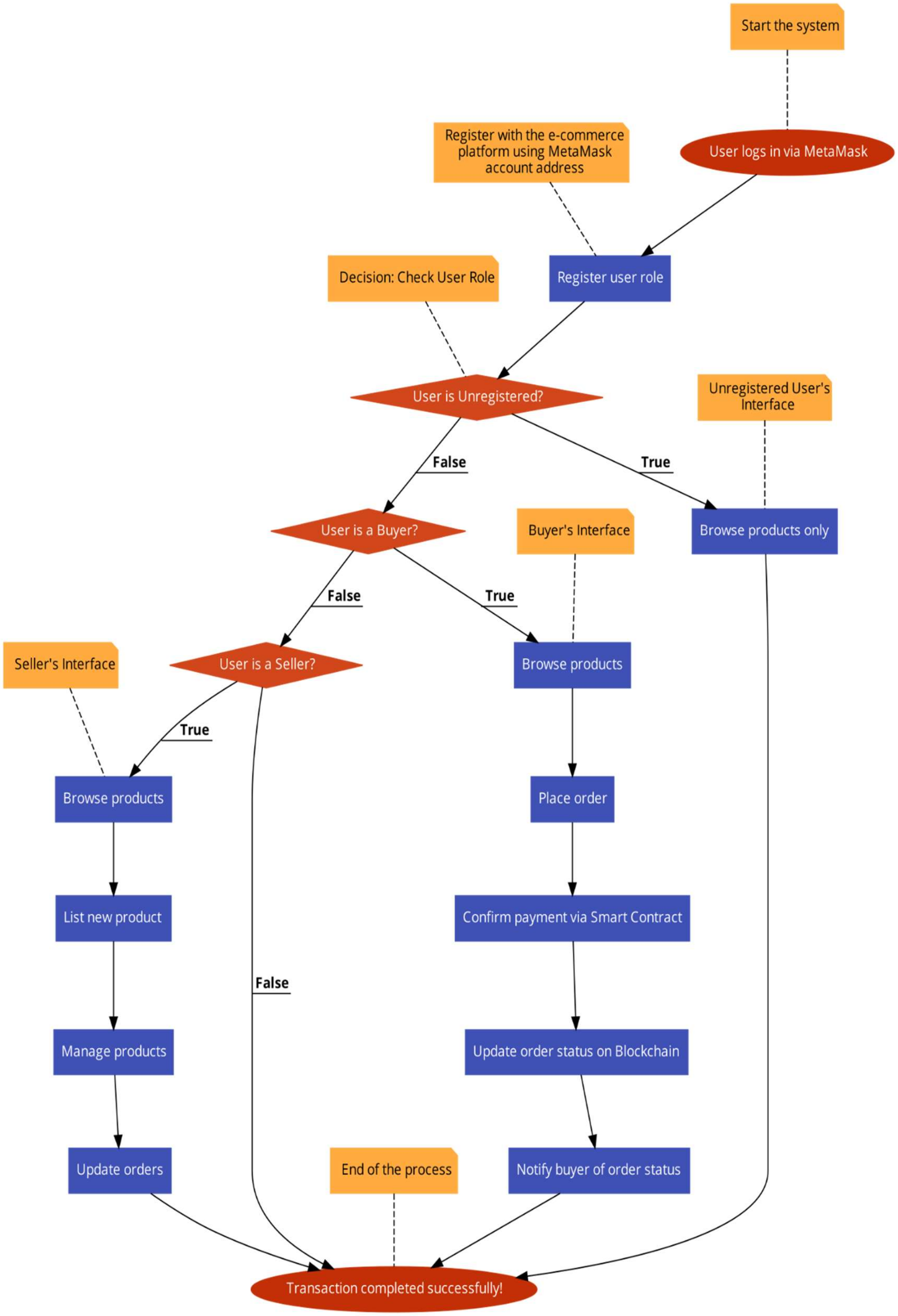

Start the system
Register with the e-commerce platform using MetaMask account address
User logs in via MetaMask
Decision: Check User Role
Register user role
User is Unregistered?
Unregistered User's Interface
False
True
User is a Buyer?
Buyer's Interface
Browse products only
False
True
Seller's Interface
User is a Seller?
Browse products
True
Browse products
Place order
List new product
Confirm payment via Smart Contract
False
Manage products
Update order status on Blockchain
Update orders
End of the process
Notify buyer of order status
Transaction completed successfully!

Figure 3-2-1-1 Top-Down System Flowchart

The top-down system flowchart shows a top-down overview of the user interaction within the blockchain e-commerce platform. In the beginning, the user logs in via

MetaMask extension and register at the platform with the MetaMask account address. After being registered, the system will check the user's role (unregistered, buyer, seller). The system will present the user with an interface relevant to the user's role.

Unregistered users can only browse products without further interaction such as placing orders. The buyers can browse the products, place orders, and make payment using smart contracts and MetaMask. The sellers can browse the products and orders placed, list a product, and manage their listed products such as manage inventory.

The flowchart ends with a transaction completed successfully.

3.2.2 Top-down System Design Block Diagram

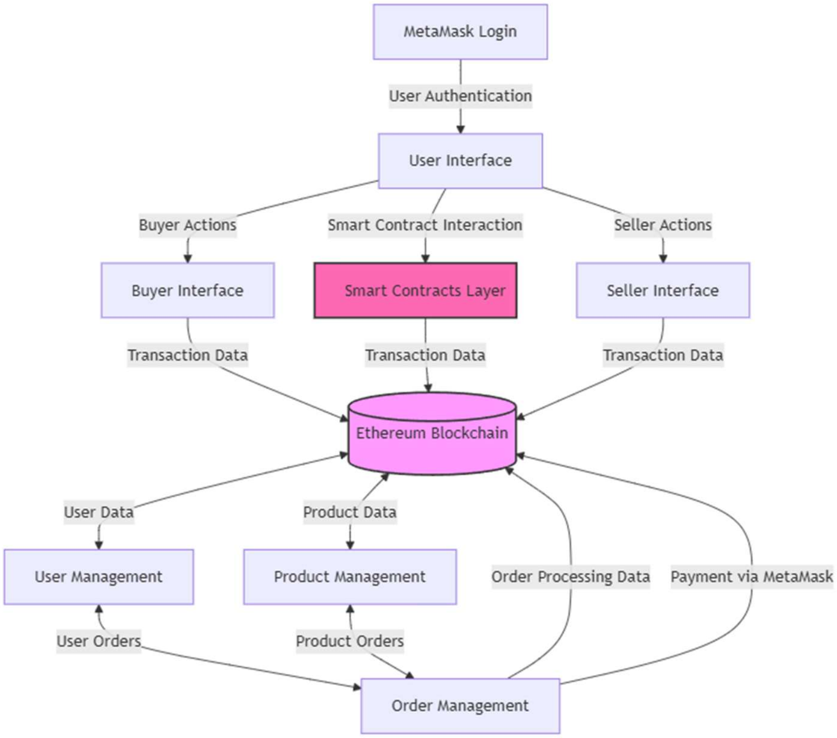


Figure 3-2-2-1 Top-Down System Design Block Diagram

The top-down system design block diagram shows a hierarchy of the blockchain e-commerce platform. It shows how different layers and components interact in facilitating user and system processes.

The first block represents the very first step of authentication via MetaMask, a popular Ethereum wallet. Users can interact with the user interface of the platform after the successful login via MetaMask.

The User Interface layer is the major interaction point for the users.

The Buyer Interface will allow users with the role of buyer to perform actions such as browsing products and placing orders.

The Seller Interface will allow users with the role of seller to perform actions such as listing their products and processing orders placed by a buyer.

The Smart Contract layer contains the core functional logics of the system. It processes buyer and seller actions and other activities before recording the transaction data on the blockchain.

The Ethereum Blockchain is a decentralized block that stores information related to transactions details along with user information, product details, and order records.

The User Management smart contract will manage information about users, including user registration data and user role.

The Product Management smart contract will manage the products and its information, such as product listing, in stock quantity, price and other.

The Order Management smart contract will manage the processing of orders, confirmation of payment via MetaMask, tracking, and status changes of orders. This interaction will ensure that payment processing is done securely over the Ethereum blockchain using MetaMask to handle user authentication and transaction signing.

3.2.3 Blockchain Overview and Structure

Blockchain technology is a digital ledger where decentralized and distributed transactions are recorded across many computers in such a manner that the users cannot go back in time and try to modify the previous transaction. A digital ledger acts like a digital notebook that keeps a permanent record of all transactions or activities, and it is shared with all users in the system so no one can secretly change or erase any of the past records. This way, security, transparency, and data integrity are ensured.

In blockchain, the data is stored in “blocks”. Each block contains information such as transaction details, the parties involved, timestamps, and other relevant data. These blocks are then linked together in a “chain” using cryptographic hashes. All these linked blocks form a “blockchain”.

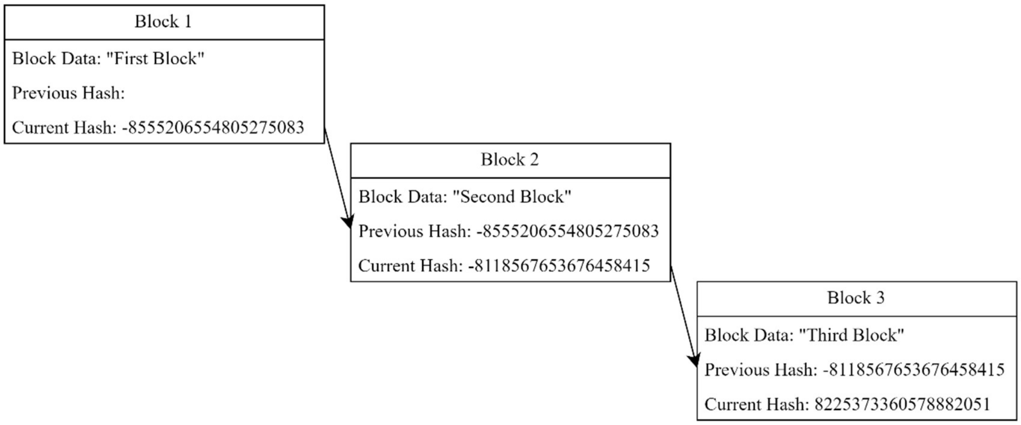

Block 1
Block Data: "First Block"
Previous Hash:
Current Hash: -8555206554805275083
Block 2
Block Data: "Second Block"
Previous Hash: -8555206554805275083
Current Hash: -8118567653676458415
Block 3
Block Data: "Third Block"
Previous Hash: -8118567653676458415
Current Hash: 8225373360578882051

Figure 3-2-3-1 Illustration of Blockchain Concept I

This diagram illustrates the simple concept of blockchain. The block will generate a “Block Hash” based on the “Block Data”, which could be the transaction details. The next chained block will generate a “Block Hash” based on the “Block Data” and “Previous Block Hash”, and the chain goes on.

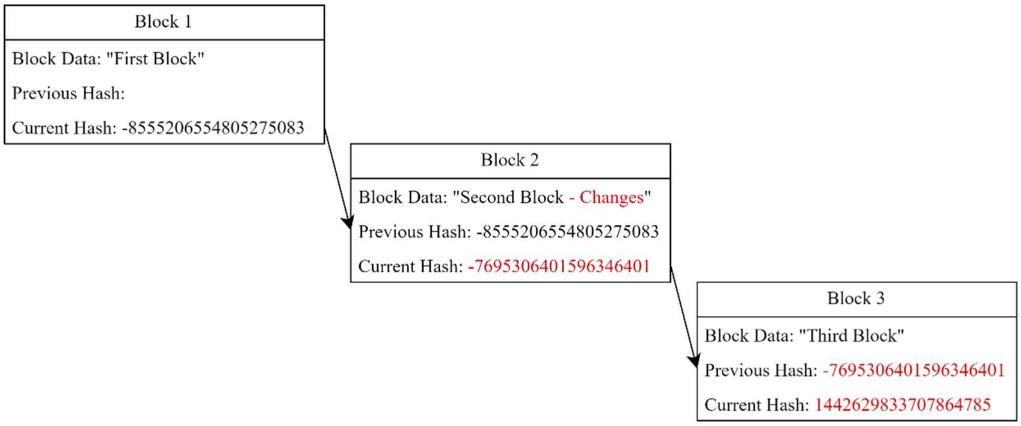
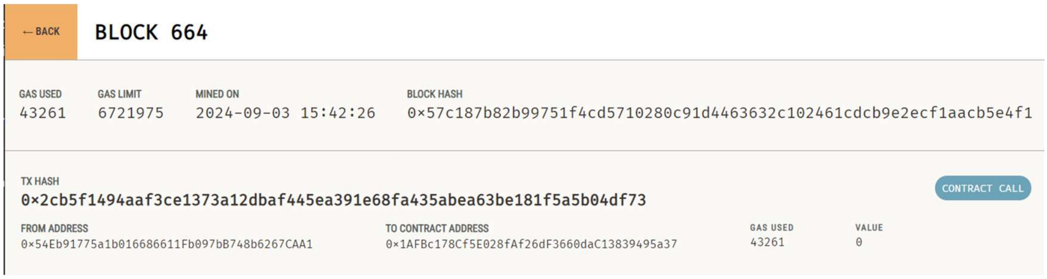

Block 1
Block Data: "First Block"
Previous Hash:
Current Hash: -8555206554805275083
Block 2
Block Data: "Second Block - Changes"
Previous Hash: -8555206554805275083
Current Hash: -7695306401596346401
Block 3
Block Data: "Third Block"
Previous Hash: -7695306401596346401
Current Hash: 1442629833707864785

Figure 3-2-3-2 Illustration of Blockchain Concept II

This diagram illustrates how even a small change in one block of the blockchain can impact the entire chain. In the second block, the data is changed from "Second Block" to "Second Block – Changes." Although the first block remains the same, the modification in Block 2 causes its unique hash to change.

This change doesn't just affect Block 2. Since Block 3 relies on the hash of Block 2 to maintain its integrity, Block 3's hash also changes. This creates a "Domino Effect" where any modification in one block affects all the following blocks in the chain.

Every node or every computer in the blockchain network has a copy of the blockchain. So, in the event of something like this happening, it is rather detectable because the changed chain will have different hashes than what's on the other nodes. This is how blockchains ensure security and tamper-proofing in general.



BLOCK 664

GAS USED 43261
GAS LIMIT 6721975
MINED ON 2024-09-03 15:42:26
BLOCK HASH 0×57c187b82b99751f4cd5710280c91d4463632c102461cdcb9e2ecf1aacb5e4f1

TX HASH 0×2cb5f1494aaf3ce1373a12dbaf445ea391e68fa435abea63be181f5a5b04df73
FROM ADDRESS 0×54Eb91775a1b016686611Fb097bB748b6267CAA1
TO CONTRACT ADDRESS 0×1AFBc178Cf5E028fAf26dF3660daC13839495a37
GAS USED 43261
VALUE 0

Figure 3-2-3-3 Screenshot of Block Information in Ganache

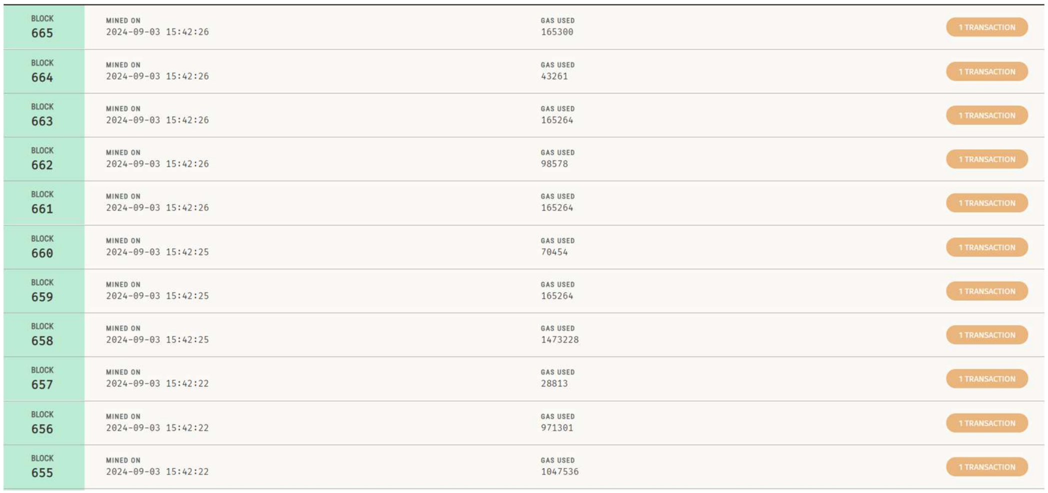
This screenshot of Ganache interface shows an example of a block on a blockchain.

Component	Description
BLOCK 664	This is the 664th block in the chain.
GAS	Gas is the fee to perform a transaction or execute a contract on the Ethereum blockchain platform.
GAS USED	The amount of computational effort used by the transactions in this block.
GAS LIMIT	The maximus gas allowed for the block.
MINED ON	The date and time when the block was mined/added to the blockchain.

BLOCK HASH	A unique identifier for this block.
TX HASH	Transaction Hash. A unique identifier for a transaction within this block.
FROM ADDRESS	The Ethereum address that originated the transaction.
TO CONTRACT ADDRESS	The smart contract address where the transaction is sent to.
GAS USED	The individual amount of gas used specifically for this transaction.
VALUE	The value transferred in this transaction. (Usually 0 for contract calls)

Table 3-2-3-1 Components of Block Data

This table describes the data stored inside a block. And there are many blocks linked together in a block chain.



BLOCK	MINED ON	GAS USED	
665	2024-09-03 15:42:26	165300	1 TRANSACTION
664	2024-09-03 15:42:26	43261	1 TRANSACTION
663	2024-09-03 15:42:26	165264	1 TRANSACTION
662	2024-09-03 15:42:26	98578	1 TRANSACTION
661	2024-09-03 15:42:26	165264	1 TRANSACTION
660	2024-09-03 15:42:25	70454	1 TRANSACTION
659	2024-09-03 15:42:25	165264	1 TRANSACTION
658	2024-09-03 15:42:25	1473228	1 TRANSACTION
657	2024-09-03 15:42:22	28813	1 TRANSACTION
656	2024-09-03 15:42:22	971301	1 TRANSACTION
655	2024-09-03 15:42:22	1047536	1 TRANSACTION

Figure 3-2-3-4 Screenshot of Many Chained Blocks in Ganache

3.3 Implementation Issues and Challenges

3.3.1 Issues and Challenges in Implementation

Implementation of the e-commerce website on blockchain was quite challenging, especially while setting up the development environment. Initial configuration of the blockchain environment was very difficult because there were few updated resources or tutorials available. Most of the tutorials found on the internet for setting up the environment were outdated and could no longer be applied, which made the process even complicated. Besides, the set-up procedures were different for the MacOS, Windows, and Linux operating systems. This has also added some complication. All these issues have made the initial phase of development really confusing and very time-consuming.

Additionally, there is another challenge which was the steep learning curve for the language of programming called Solidity. It is mandatory for creating smart contracts running on the Ethereum blockchain network. The blockchain itself was also a completely new domain for the developer, which required months of study to understand the principles and their application in this project. This foundational knowledge was important for ensuring the integrity and functionality of the blockchain-based system. On the other hand, integration of the backend blockchain with the front-end interface posted difficulties. It was quite challenging to ensure smooth communication between the front-end user interface and the blockchain, especially when handling smart contract interactions.

3.3.2 Novel Aspects

The novelty of this project is in using blockchain technology to remove middlemen in an e-commerce process. This helps reduce the transaction costs. At the same time, it also enhances the trust and transparency between buyer and seller. By using smart contracts, the processes such as order management, product management, and user authentication are automated in a tamper-proof manner in the decentralized e-commerce environment. Furthermore, integration with MetaMask enhances security and allows users fully control their transactions, setting a new standard for secure and decentralized online marketplaces.

3.4 Timeline

3.4.1 Current Semester Plan

Below is the Gantt chart for the FYP1 project timeline.

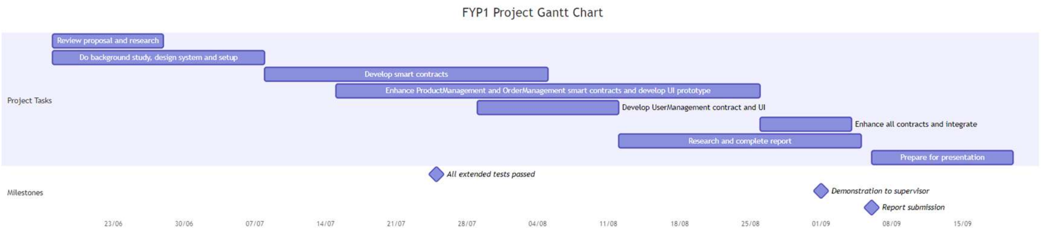


Figure 3-4-1-1 FYP1 Project Gantt Chart

Starting from week 1, the previous proposal was reviewed to check if any changes can be made to the system. The background study was done to have a comprehensive insight into blockchain technology, followed by the System Design and the Environment Set Up finished on Week 3. The contracts development for User, Product, and Order was developed on Week 4.

Frontend Prototype add products and place orders was set up on Week 5. In Week 6, a milestone was achieved where the extended tests for "ProductManagement" and "OrderManagement" passed. The "UserManagement" contract was developed and enhanced, then integrated into UI. Meanwhile, the research and work on the report was started from Week 9 onwards. In Week 11, all smart contracts were reviewed and enhanced, before integration into UI. A demonstration of the prototype was reviewed by supervisor. The report was submitted on due date.

3.4.2 Next Semester Plan

In next FYP2 semester, the development of the project will be divided into three phases, which are early phase, mid phase, and final phase.

In early phase, the main objective includes refinement of the system. The work done during FYP1 will be reviewed. Feedback from supervisor and users will be collected and make improvements. The key milestone in this phase could be finishing the core system improvement.

In the mid phase, the main objective includes extending and integrating the features of the system. Advanced functions like advanced product management will be developed and integrated. A full system integration will be done to ensure all components from backend to frontend are integrated seamlessly. The key milestones in this phase include completing the integration of the system.

In the final phase, the main objective will be conducting final testing and creating documentation. A full-scale testing of the whole system for functionality, integration, performance, and security will be performed. Detailed documentation of the whole project including technical specification, user manuals, and system architecture will be created. Testing process and results are also documented in this stage.

This is a preliminary plan that may be modified according to the development of the project, feedback from the supervisors, and issues that arise in the process. It will be updated if necessary for the project.

CHAPTER 4

Preliminary Work

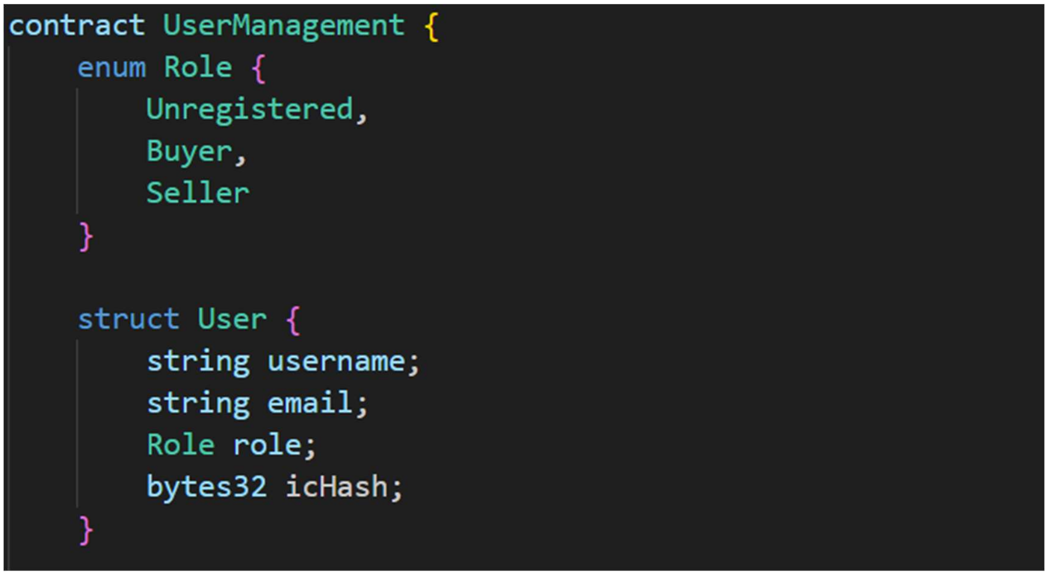
4.1 Overview

This chapter presents the work done based on the methodologies outlined in Chapter 3. This section focuses on proving the feasibility of the proposed blockchain-based e-commerce platform through the performance of preliminary tests and implementations.

4.2 Preliminary Implementation

For the preparatory work, setup tasks for the development environment of blockchain including the installation and configuration processes for Truffle, Ganache, and MetaMask were done. With a system design in hand, the next step was the development of smart contracts for managing users, products, and orders using the language of Solidity. These contracts were purposed to handle the critical operations: user registration, product listing, and order processing. Test deployment of these contracts was performed in a locally created blockchain environment by Truffle known as Ganache. Contracts ran smoothly, performing targeted functionalities.

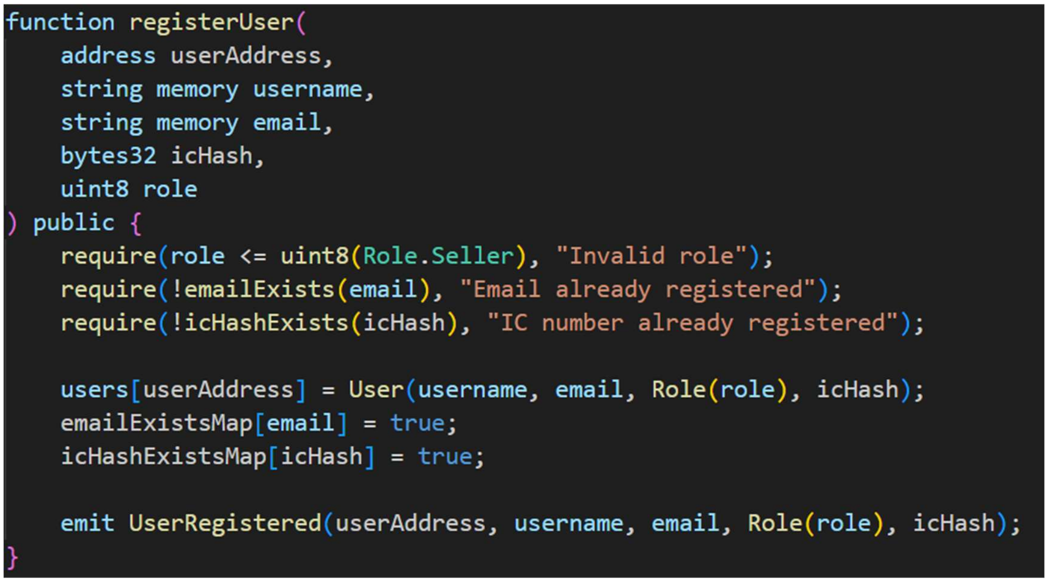
Below are the code snippets.



```
contract UserManagement {
    enum Role {
        Unregistered,
        Buyer,
        Seller
    }

    struct User {
        string username;
        string email;
        Role role;
        bytes32 icHash;
    }
```

Figure 4-2-1 Code Snippet of UserManagement Smart Contract

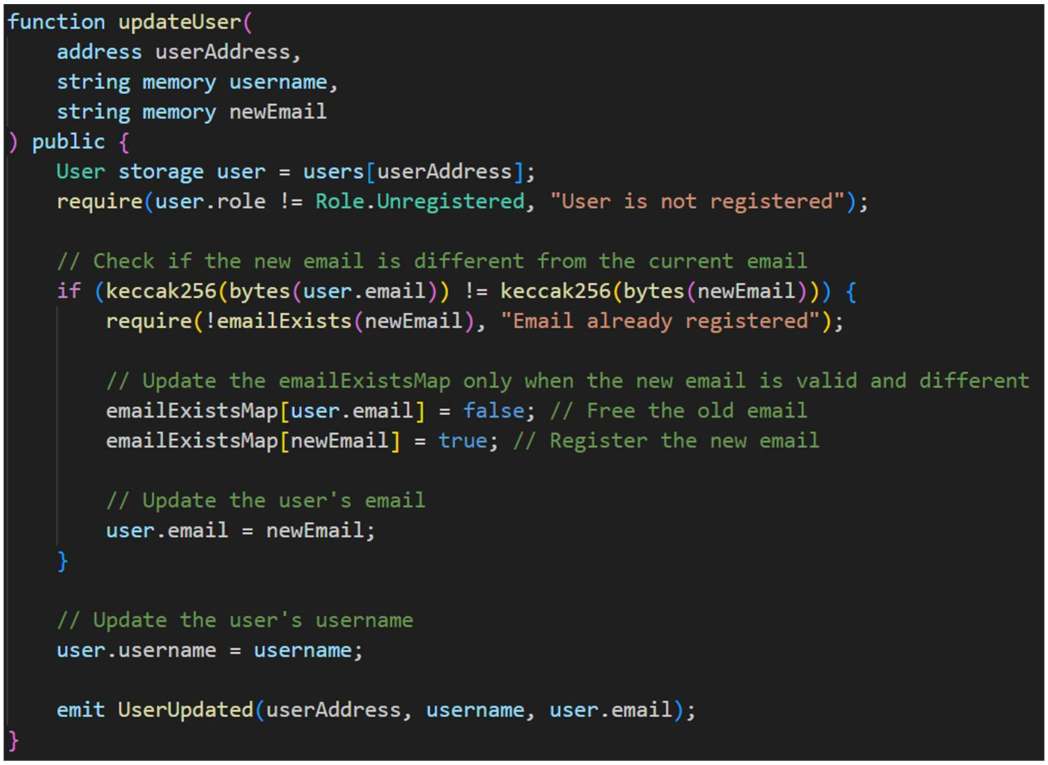
The code snippet defines a UserManagement contract in Solidity with a Role enum for user roles (Unregistered, Buyer, Seller) and a User struct that stores a user's username, email, role, and a hashed IC Number “icHash” value for identification.



```
function registerUser(
    address userAddress,
    string memory username,
    string memory email,
    bytes32 icHash,
    uint8 role
) public {
    require(role <= uint8(Role.Seller), "Invalid role");
    require(!emailExists(email), "Email already registered");
    require(!icHashExists(icHash), "IC number already registered");

    users[userAddress] = User(username, email, Role(role), icHash);
    emailExistsMap[email] = true;
    icHashExistsMap[icHash] = true;

    emit UserRegistered(userAddress, username, email, Role(role), icHash);
}
```

Figure 4-2-2 Code Snippet of registerUser Function

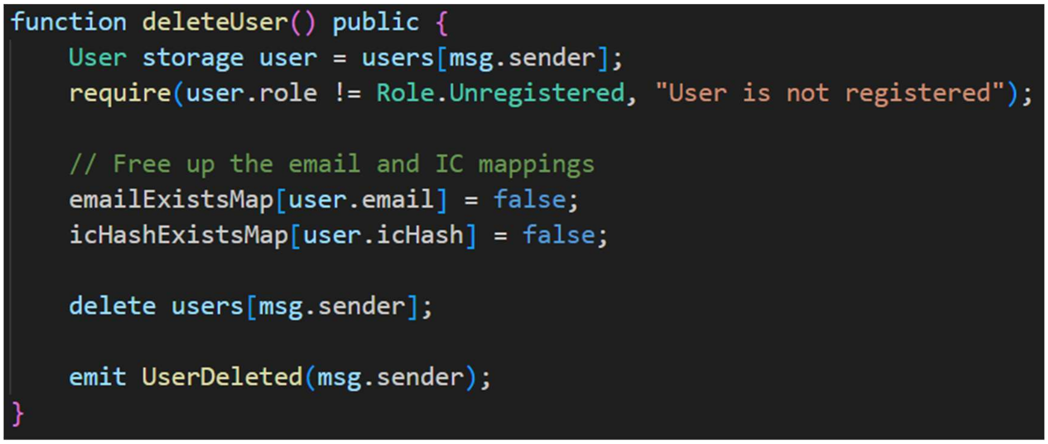
This “registerUser” function allows a user to register by providing their address, username, email, “icHash”, and role. It ensures the role is valid and checks if the email and IC number have already been registered. If all checks pass, it stores the user's details and emits a “UserRegistered” event to log the registration.



```
function updateUser(
    address userAddress,
    string memory username,
    string memory newEmail
) public {
    User storage user = users[userAddress];
    require(user.role != Role.Unregistered, "User is not registered");

    // Check if the new email is different from the current email
    if (keccak256(bytes(user.email)) != keccak256(bytes(newEmail))) {
        require(!emailExists(newEmail), "Email already registered");

        // Update the emailExistsMap only when the new email is valid and different
        emailExistsMap[user.email] = false; // Free the old email
        emailExistsMap[newEmail] = true; // Register the new email

        // Update the user's email
        user.email = newEmail;
    }

    // Update the user's username
    user.username = username;

    emit UserUpdated(userAddress, username, user.email);
}
```

Figure 4-2-3 Code Snippet of updateUser Function

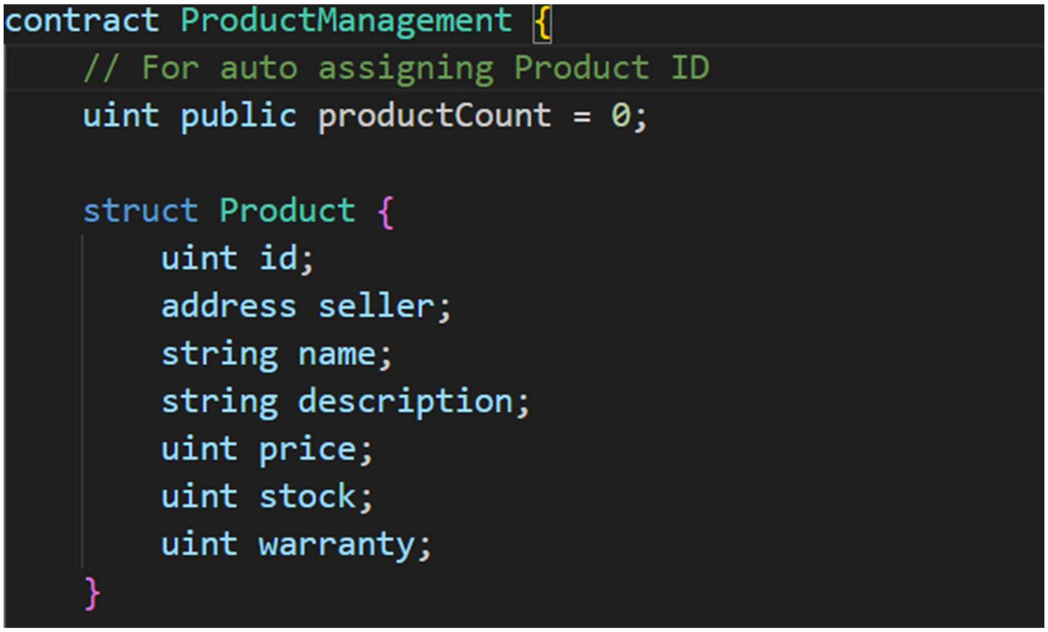
This “updateUser” function allows a registered user to update their username and email. It first verifies that the user is registered. If the new email is different from the existing one, it checks if the new email is already registered, updates the email mapping, and then updates the user's email and username. A “UserUpdated” event is emitted to record the update.



```
function deleteUser() public {
    User storage user = users[msg.sender];
    require(user.role != Role.Unregistered, "User is not registered");

    // Free up the email and IC mappings
    emailExistsMap[user.email] = false;
    icHashExistsMap[user.icHash] = false;

    delete users[msg.sender];

    emit UserDeleted(msg.sender);
}
```

Figure 4-2-4 Code Snippet of deleteUser Function

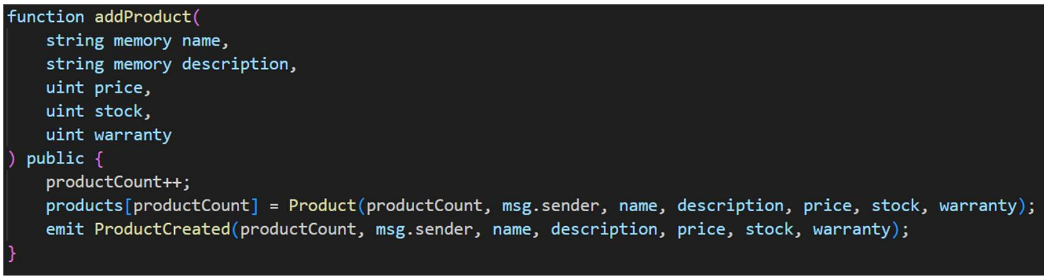
This function lets a user delete their account by first ensuring they are registered. It then frees up the email and "icHash" mappings associated with the user before deleting the user's data from the contract. The function emits a "UserDeleted" event to confirm the deletion. However, this function is not yet integrated in the front-end due to technical issue.



```
contract ProductManagement {
    // For auto assigning Product ID
    uint public productCount = 0;

    struct Product {
        uint id;
        address seller;
        string name;
        string description;
        uint price;
        uint stock;
        uint warranty;
    }
```

Figure 4-2-5 Code Snippet of ProductManagement Smart Contract

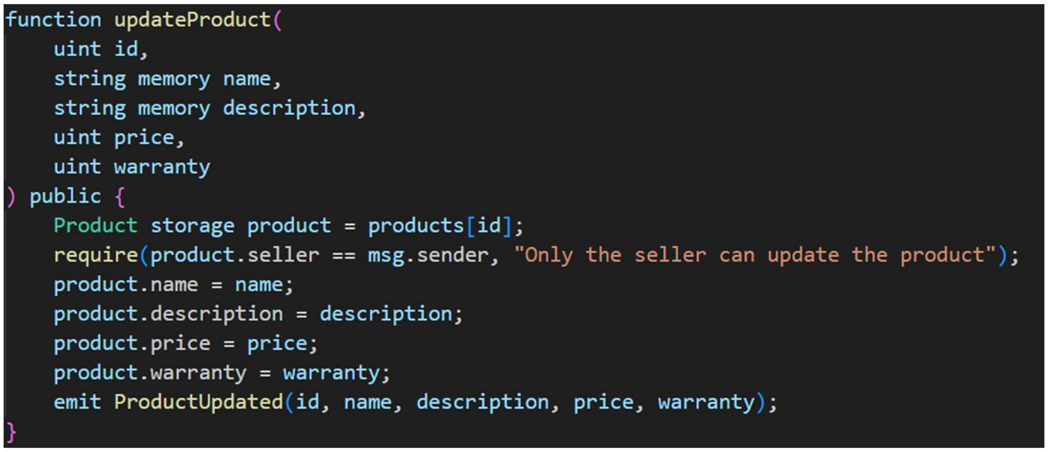
The code snippet defines a "ProductManagement" contract in Solidity which contains a "productCount" variable for automatically assigning product IDs and a Product struct that holds information about each product, including its ID, seller, name, description, price, stock, and warranty.



```
function addProduct(
    string memory name,
    string memory description,
    uint price,
    uint stock,
    uint warranty
) public {
    productCount++;
    products[productCount] = Product(productCount, msg.sender, name, description, price, stock, warranty);
    emit ProductCreated(productCount, msg.sender, name, description, price, stock, warranty);
}
```

Figure 4-2-6 Code Snippet of addProduct Function

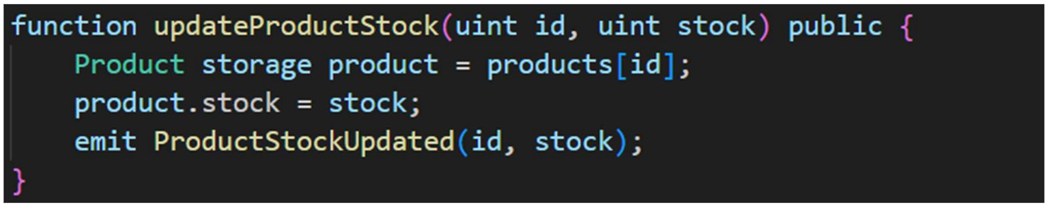
This “addProduct” function allows a seller to add a new product by providing its name, description, price, stock, and warranty. The function automatically increments the “productCount”, assigns a new product ID, and stores the product details in the products mapping. It then emits a “ProductCreated” event to log the creation of the new product.



```
function updateProduct(
    uint id,
    string memory name,
    string memory description,
    uint price,
    uint warranty
) public {
    Product storage product = products[id];
    require(product.seller == msg.sender, "Only the seller can update the product");
    product.name = name;
    product.description = description;
    product.price = price;
    product.warranty = warranty;
    emit ProductUpdated(id, name, description, price, warranty);
}
```

Figure 4-2-7 Code Snippet of updateProduct Function

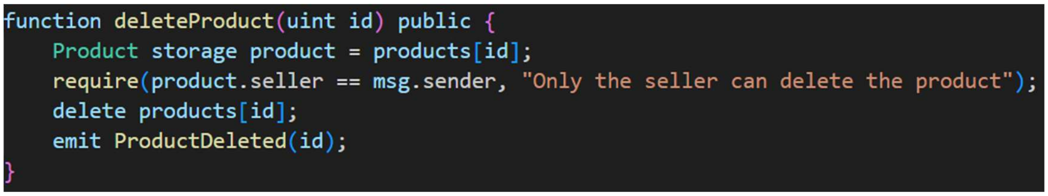
This “updateProduct” function allows a seller to update the details of an existing product, such as its name, description, price, and warranty. It first verifies that the fucntion caller is the seller of the product, then updates the relevant fields, and finally emits a “ProductUpdated” event to log the changes. However, this function is not yet integrated in the front-end due to technical issue.



```
function updateProductStock(uint id, uint stock) public {
    Product storage product = products[id];
    product.stock = stock;
    emit ProductStockUpdated(id, stock);
}
```

Figure 4-2-8 Code Snippet of updateProductStock Function

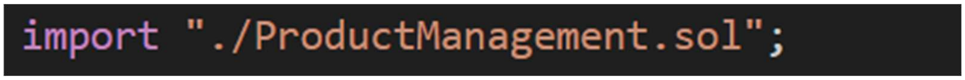
This “updateProductStock” function enables a seller or the system to update the stock quantity of an existing product. It retrieves the product from the products mapping using its ID, updates the stock value, and emits a “ProductStockUpdated” event to record the change in stock.



```
function deleteProduct(uint id) public {
    Product storage product = products[id];
    require(product.seller == msg.sender, "Only the seller can delete the product");
    delete products[id];
    emit ProductDeleted(id);
}
```

Figure 4-2-9 Code Snippet of deleteProduct Function

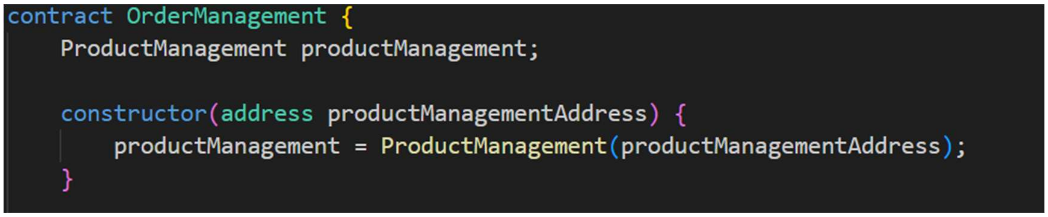
This “deleteProduct” function allows a seller to delete a product they own by specifying its ID. It checks that the function caller is the seller of the product, deletes the product from the products mapping, and emits a “ProductDeleted” event to log the deletion.



```
import "./ProductManagement.sol";
```

Figure 4-2-10 Code Snippet of Importing ProductManagement in OrderManagement

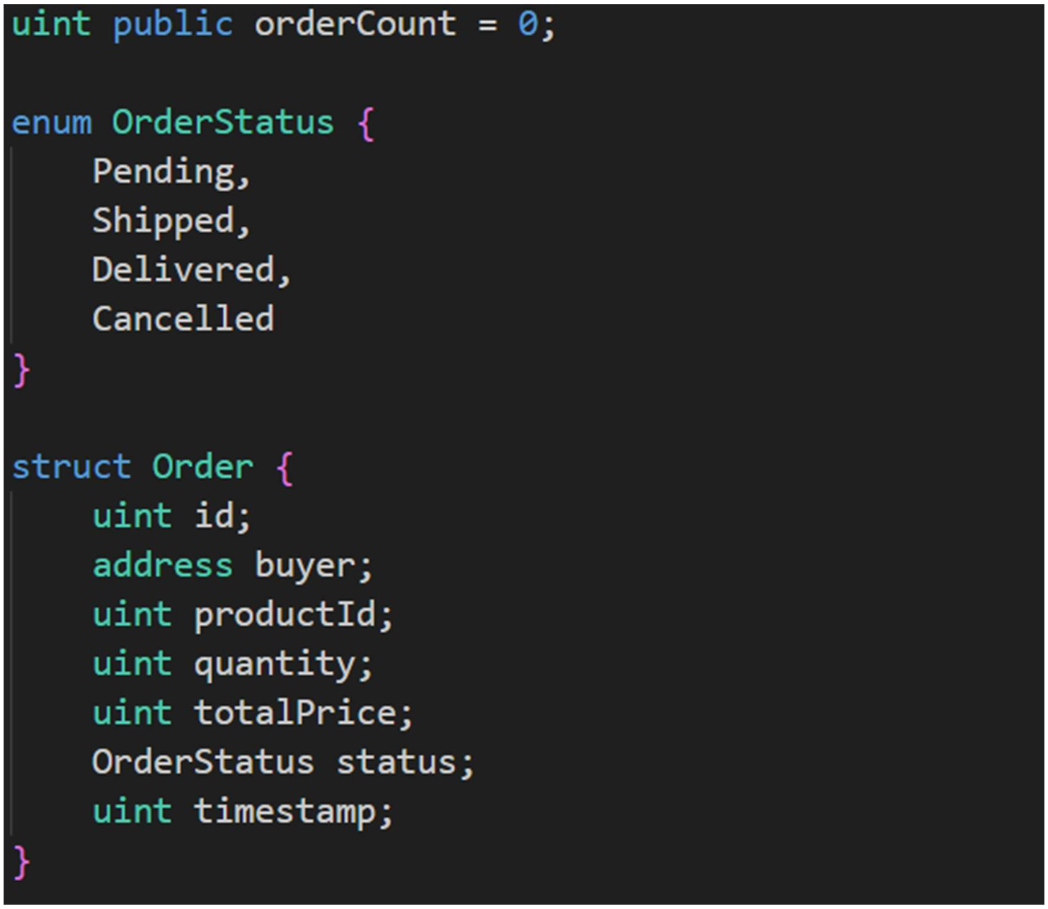
This line of code imports the “ProductManagement.sol”, which allows the “OrderManagement” contract to access and interact with the functions and data structures defined in the “ProductManagement” contract.



```
contract OrderManagement {
    ProductManagement productManagement;

    constructor(address productManagementAddress) {
        productManagement = ProductManagement(productManagementAddress);
    }
```

Figure 4-2-11 Code Snippet of OrderManagement Smart Contract

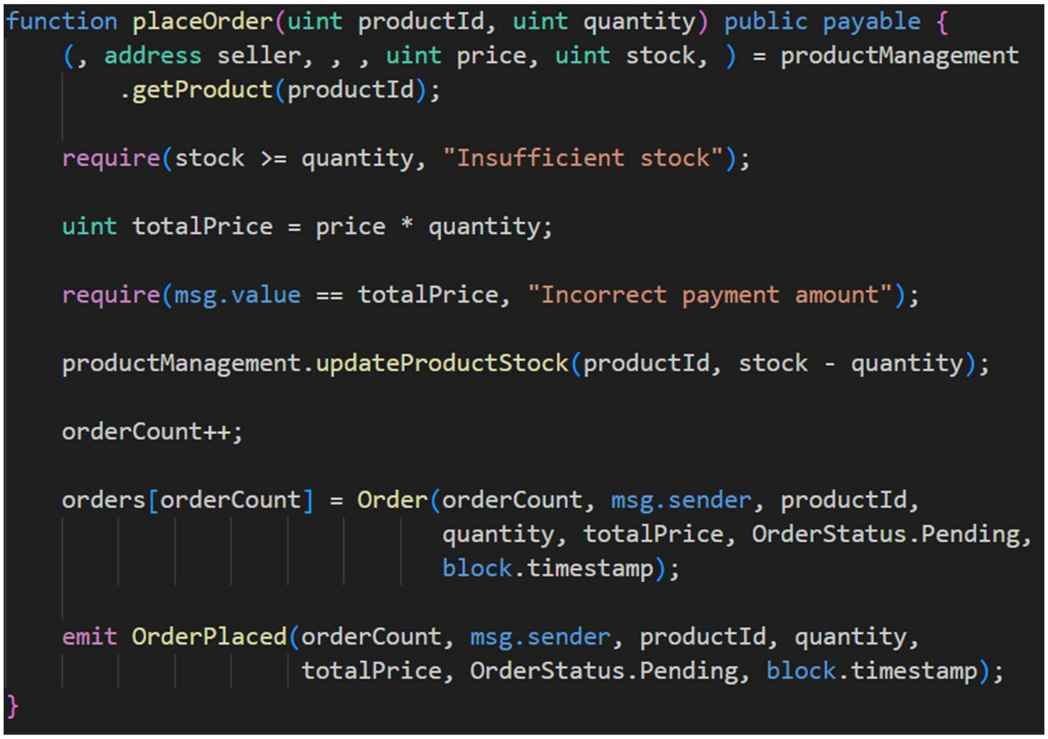
The code snippet defines a "ProductManagement" instance to interact with the product management system and a constructor that initializes this instance by taking the address of the deployed "ProductManagement" contract as a parameter.



```
uint public orderCount = 0;

enum OrderStatus {
    Pending,
    Shipped,
    Delivered,
    Cancelled
}

struct Order {
    uint id;
    address buyer;
    uint productId;
    uint quantity;
    uint totalPrice;
    OrderStatus status;
    uint timestamp;
}
```

Figure 4-2-12 Code Snippet of Head Lines of OrderManagement

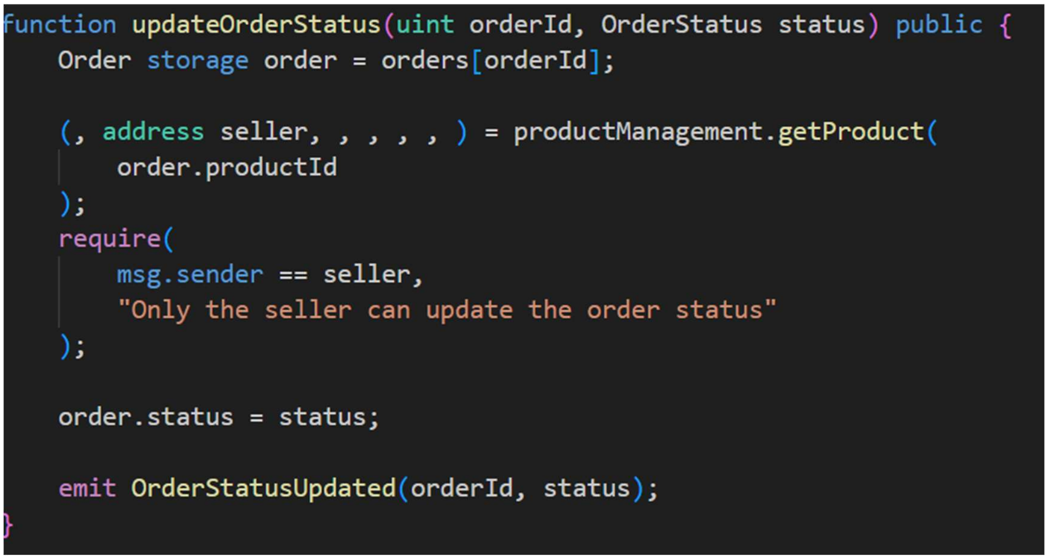
This code snippet defines a "orderCount" variable for automatically assigning order IDs and the definition of an "OrderStatus" enum, which includes possible statuses an order can have (Pending, Shipped, Delivered, Cancelled). It also defines an Order struct that holds essential information about an order, including its ID, buyer's address, product ID, quantity, total price, status, and timestamp.



```
function placeOrder(uint productId, uint quantity) public payable {
    (, address seller, , , uint price, uint stock, ) = productManagement
        .getProduct(productId);

    require(stock >= quantity, "Insufficient stock");

    uint totalPrice = price * quantity;

    require(msg.value == totalPrice, "Incorrect payment amount");

    productManagement.updateProductStock(productId, stock - quantity);

    orderCount++;

    orders[orderCount] = Order(orderCount, msg.sender, productId,
                               quantity, totalPrice, OrderStatus.Pending,
                               block.timestamp);

    emit OrderPlaced(orderCount, msg.sender, productId, quantity,
                     totalPrice, OrderStatus.Pending, block.timestamp);
}
```

Figure 4-2-13 Code Snippet of placeOrder Function

This "placeOrder" function allows a buyer to place an order for a product by specifying the product ID and quantity. It first retrieves the product details and checks for sufficient stock and correct payment. Then it updates the product stock, creates a new order, increments the "orderCount", and emits an "OrderPlaced" event to log the order details.

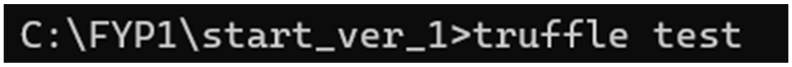


```
function updateOrderStatus(uint orderId, OrderStatus status) public {
    Order storage order = orders[orderId];

    (, address seller, , , , , ) = productManagement.getProduct(
        order.productId
    );
    require(
        msg.sender == seller,
        "Only the seller can update the order status"
    );

    order.status = status;

    emit OrderStatusUpdated(orderId, status);
}
```

Figure 4-2-14 Code Snippet of updateOrderStatus Function

This “updateOrderStatus” function allows the seller to update the status of an existing order. It first ensures that only the seller of the product can perform this action, then updates the order status accordingly, and emits an “OrderStatusUpdated” event to log the status change.

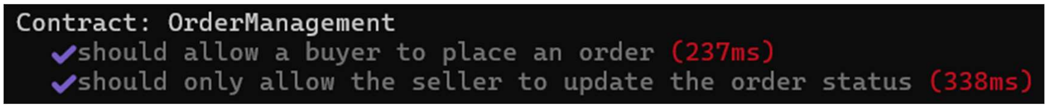
4.3 Preliminary Results



```
C:\FYP1\start_ver_1>truffle test
```

Figure 4-3-1 Command to Run Test on Smart Contracts

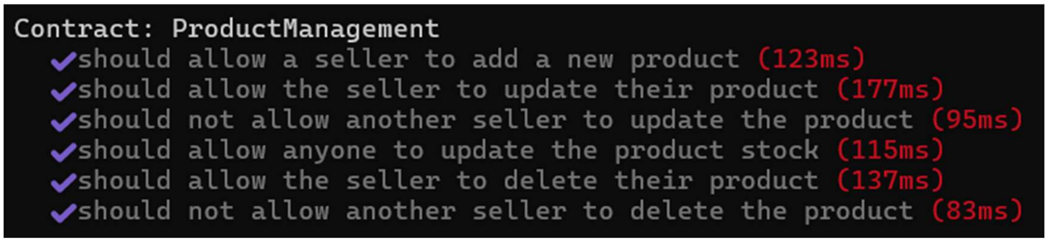
Running command in Command Prompt to test the contracts.



```
Contract: OrderManagement
   ✓should allow a buyer to place an order (237ms)
   ✓should only allow the seller to update the order status (338ms)
```

Figure 4-3-2 Test Result of OrderManagement Smart Contract

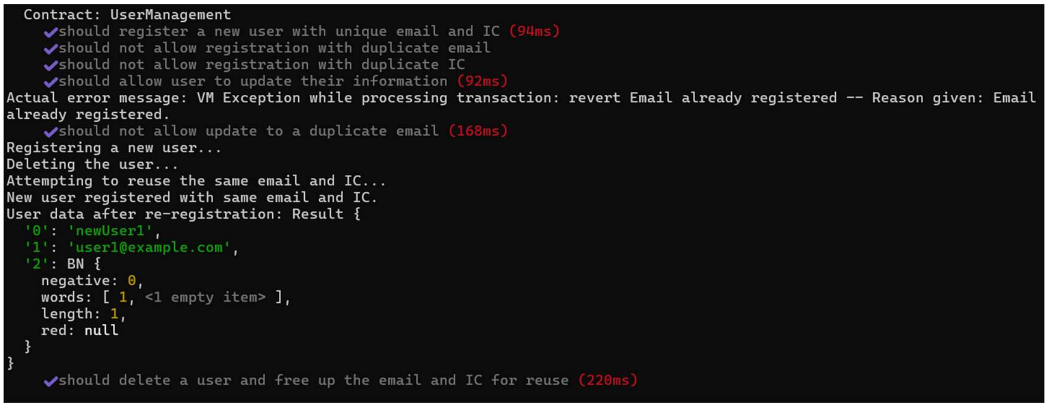
The test result shows the “OrderManagement” smart contract has passed two major functionalities, which allows a buyer to place an order and only allow the seller to update the order status.



```
Contract: ProductManagement
   ✓should allow a seller to add a new product (123ms)
   ✓should allow the seller to update their product (177ms)
   ✓should not allow another seller to update the product (95ms)
   ✓should allow anyone to update the product stock (115ms)
   ✓should allow the seller to delete their product (137ms)
   ✓should not allow another seller to delete the product (83ms)
```

Figure 4-3-3 Test Result of ProductManagement Smart Contract

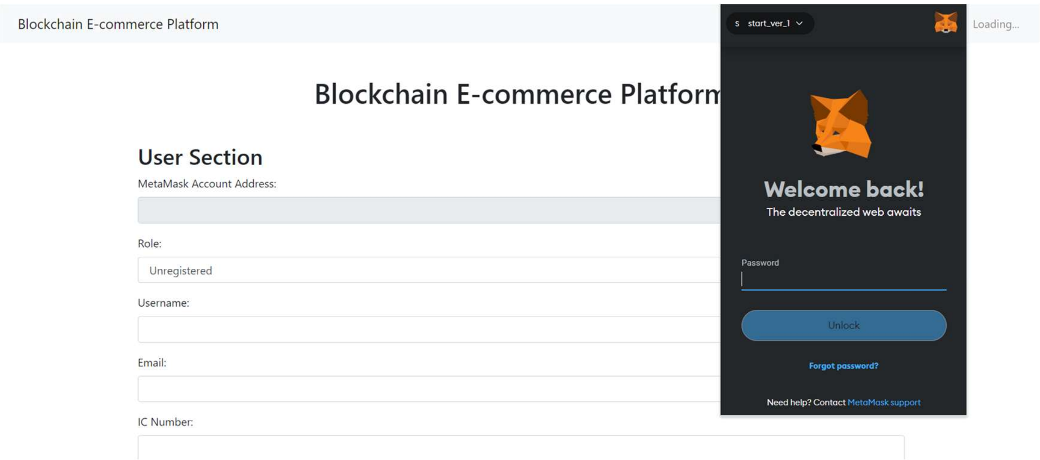
The test result shows the “ProductManagement” smart contract had all the functions work as expected. This contract allows the seller to add new products, to update their products, and to delete them. Besides that, it correctly limits other sellers from updating a product and from deleting any product they do not own. This will allow the system to update the stock of a product automatically after a deduction from order placing by a buyer.



```
  Contract: UserManagement
    ✔should register a new user with unique email and IC (94ms)
    ✔should not allow registration with duplicate email
    ✔should not allow registration with duplicate IC
    ✔should allow user to update their information (92ms)
Actual error message: VM Exception while processing transaction: revert Email already registered -- Reason given: Email
already registered.
    ✔should not allow update to a duplicate email (168ms)
Registering a new user...
Deleting the user...
Attempting to reuse the same email and IC...
New user registered with same email and IC.
User data after re-registration: Result {
  '0': 'newUser1',
  '1': 'user1@example.com',
  '2': BN {
    negative: 0,
    words: [ 1, <1 empty item> ],
    length: 1,
    red: null
  }
}
    ✔should delete a user and free up the email and IC for reuse (220ms)
```

Figure 4-3-4 Test Result of UserManagement Smart Contract

The test result shows the "UserManagement" smart contract successfully registers new users with unique email and IC number and does not allow registration for duplicate email and IC number. The updating of users' information while rejecting the update of existing email to a duplicated email is also enabled. Results from the tests also show that a user can be deleted by the contract and make the email and IC free to be reused again.



Blockchain E-commerce Platform

Loading...

Blockchain E-commerce Platform

User Section

MetaMask Account Address:

Role: Unregistered

Username:

Email:

IC Number:

start_ver_1

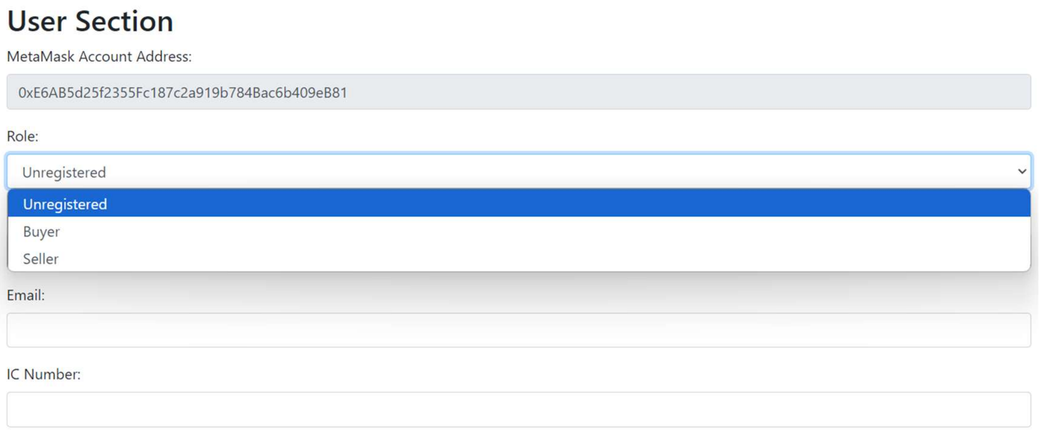
Welcome back!

The decentralized web awaits

Password

Figure 4-3-5 Screenshot of MetaMask Extension

For the front-end, the user will have log in to a MetaMask account before using the platform.



User Section

MetaMask Account Address: 0xE6AB5d25f2355Fc187c2a919b784Bac6b409eB81

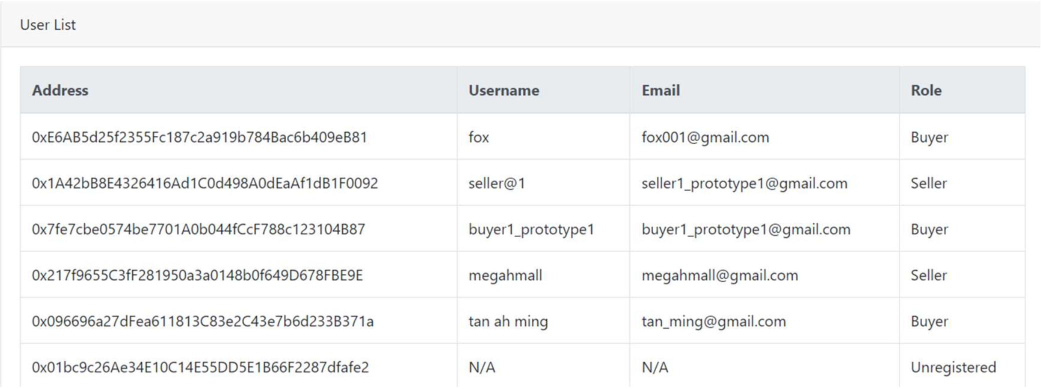
Role: Unregistered

Email:

IC Number:

Figure 4-3-6 Screenshot of User Section in Front-End Interface

The “User Section” allows users to interact with the blockchain-based platform by using the MetaMask account address and selecting a particular role of either unregistered, buyer, or seller from the dropdown menu, then followed by the username, email, and IC number.

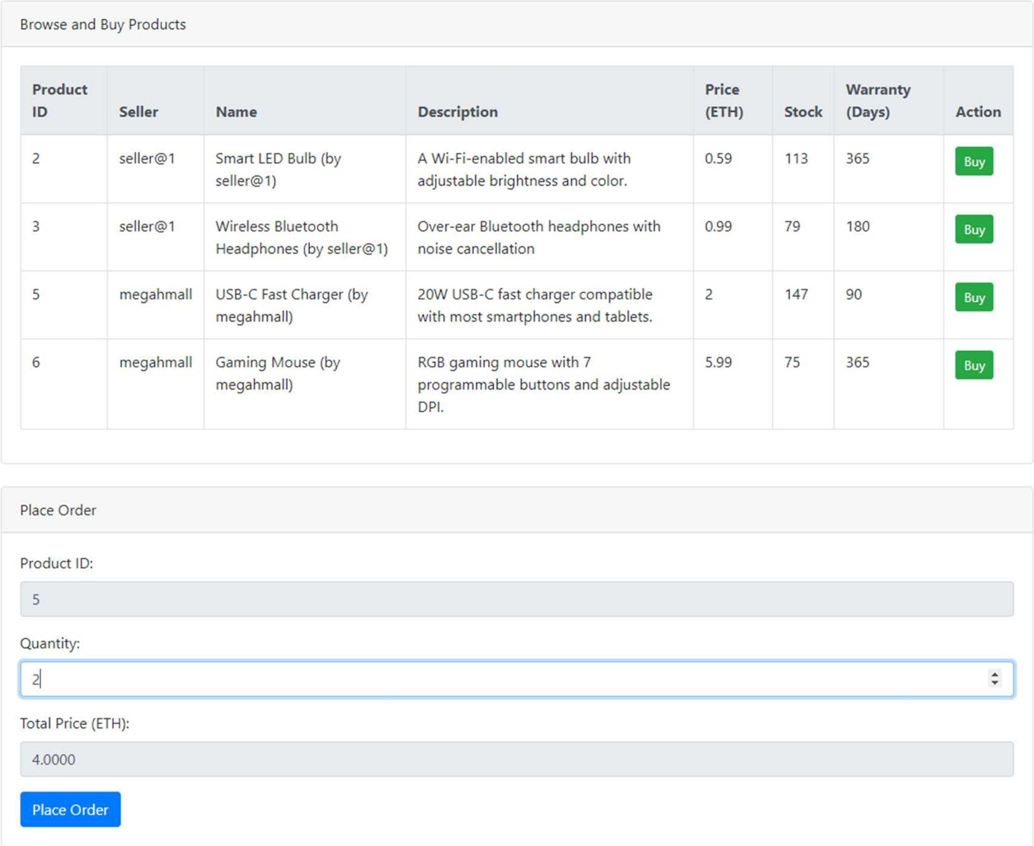


User List

Address	Username	Email	Role
0xE6AB5d25f2355Fc187c2a919b784Bac6b409eB81	fox	fox001@gmail.com	Buyer
0x1A42bB8E4326416Ad1C0d498A0dEaAf1dB1F0092	seller@1	seller1_prototype1@gmail.com	Seller
0x7fe7cbe0574be7701A0b044fCcF788c123104B87	buyer1_prototype1	buyer1_prototype1@gmail.com	Buyer
0x217f9655C3fF281950a3a0148b0f649D678FBE9E	megahmall	megahmall@gmail.com	Seller
0x096696a27dFea611813C83e2C43e7b6d233B371a	tan ah ming	tan_ming@gmail.com	Buyer
0x01bc9c26Ae34E10C14E55DD5E1B66F2287dfafe2	N/A	N/A	Unregistered

Figure 4-3-7 Screenshot of User List in Front-End Interface

The “User List” is a table displaying the blockchain account address, username, email, and assigned role, which could be one of either Buyer, Seller, or Unregistered. This interface provides a total overview of all the users currently using the platform by showing who has been registered and into which role.



Browse and Buy Products

Product ID	Seller	Name	Description	Price (ETH)	Stock	Warranty (Days)	Action
2	seller@1	Smart LED Bulb (by seller@1)	A Wi-Fi-enabled smart bulb with adjustable brightness and color.	0.59	113	365	Buy
3	seller@1	Wireless Bluetooth Headphones (by seller@1)	Over-ear Bluetooth headphones with noise cancellation	0.99	79	180	Buy
5	megahmall	USB-C Fast Charger (by megahmall)	20W USB-C fast charger compatible with most smartphones and tablets.	2	147	90	Buy
6	megahmall	Gaming Mouse (by megahmall)	RGB gaming mouse with 7 programmable buttons and adjustable DPI.	5.99	75	365	Buy

Place Order

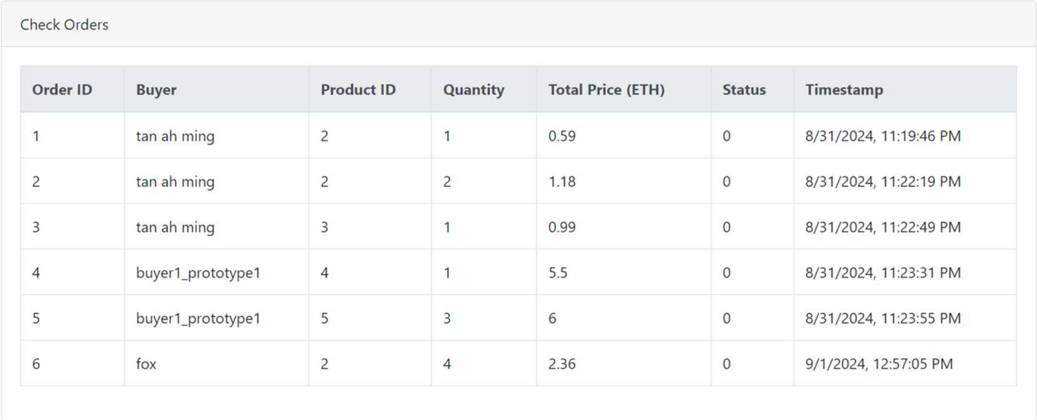
Product ID: 5

Quantity: 2

Total Price (ETH): 4.0000

Figure 4-3-8 Screenshot of Browse and Buy Section in Front-End Interface

The “Browse and Buy Products” section includes list of all available products, showing a product ID, seller, name, description, price in ETH, quantity in stock, and warranty period. The buyer can click the “Buy” button to buy a product. The product ID will be automatically filled, and the buyer can specify the quantity he wants to buy in the "Place Order" section below. The total price will be automatically calculated. Then the user will be able to place an order using the “Place Order” button.

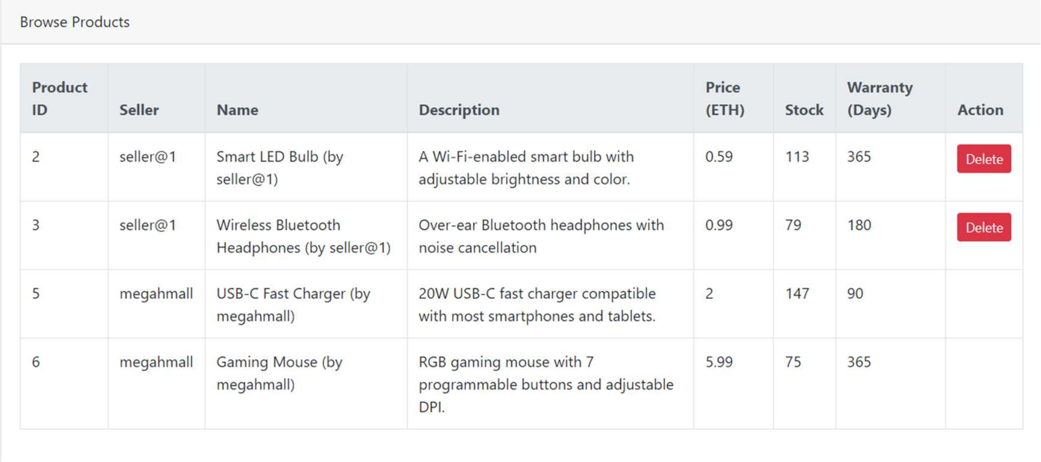


Check Orders

Order ID	Buyer	Product ID	Quantity	Total Price (ETH)	Status	Timestamp
1	tan ah ming	2	1	0.59	0	8/31/2024, 11:19:46 PM
2	tan ah ming	2	2	1.18	0	8/31/2024, 11:22:19 PM
3	tan ah ming	3	1	0.99	0	8/31/2024, 11:22:49 PM
4	buyer1_prototype1	4	1	5.5	0	8/31/2024, 11:23:31 PM
5	buyer1_prototype1	5	3	6	0	8/31/2024, 11:23:55 PM
6	fox	2	4	2.36	0	9/1/2024, 12:57:05 PM

Figure 4-3-9 Screenshot of Check Orders Section in Front-End Interface

The “Check Orders” section provides a summary of all orders made on the platform with relevant information of Order ID, Buyer name, Product ID, Quantity, Total Price in ETH, Order Status, and Timestamp when the order was placed. This table allows both users and sellers to track and manage recent transactions and their status.



Browse Products

Product ID	Seller	Name	Description	Price (ETH)	Stock	Warranty (Days)	Action
2	seller@1	Smart LED Bulb (by seller@1)	A Wi-Fi-enabled smart bulb with adjustable brightness and color.	0.59	113	365	Delete
3	seller@1	Wireless Bluetooth Headphones (by seller@1)	Over-ear Bluetooth headphones with noise cancellation	0.99	79	180	Delete
5	megahmall	USB-C Fast Charger (by megahmall)	20W USB-C fast charger compatible with most smartphones and tablets.	2	147	90	
6	megahmall	Gaming Mouse (by megahmall)	RGB gaming mouse with 7 programmable buttons and adjustable DPI.	5.99	75	365	

Figure 4-3-10 Screenshot of Browse Product Section in Front-End Interface

When the user’s role is a seller, the “Browse Products” section will list the products on sale with relevant information of Product ID, Seller, Name, Description, Price in ETH, Stock quantity, and Warranty period. It also includes a column named “Action”, containing a “Delete” button for the seller to remove his product from this list.

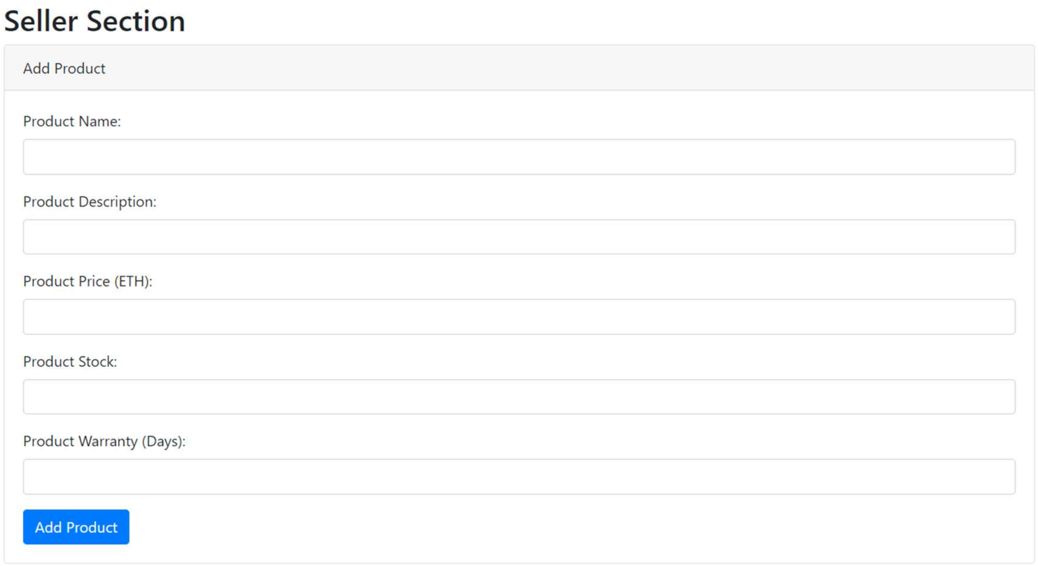


Figure 4-3-11 Screenshot of Seller Section in Front-End Interface

The “Seller Section” interface allows the seller to add new products on the platform by entering a name, description, price in ETH, quantity in stock, and warranty period in days. Once the above fields are filled, the seller can click the “Add Product” button to list his product for sale.

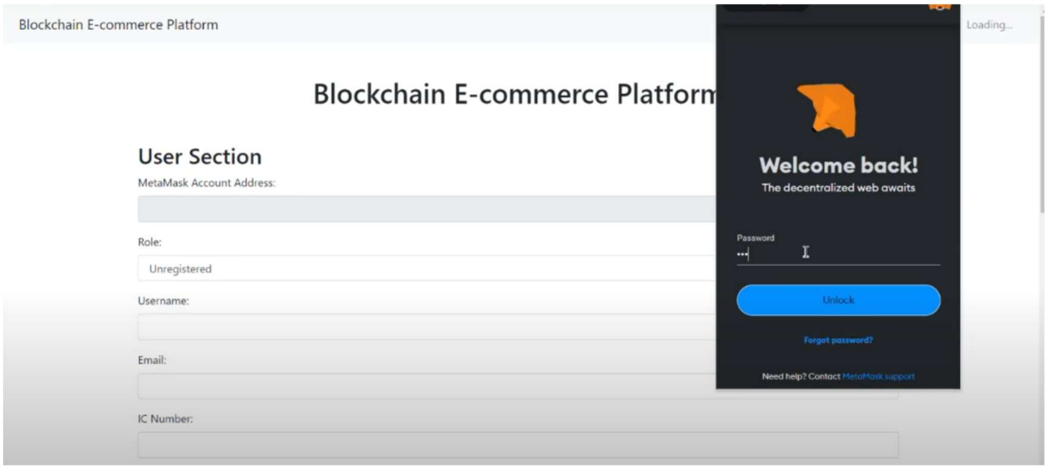
4.4 Use Case Testing Scenarios

4.4.1 User Register Scenario

A user who is already a MetaMask user and has an MetaMask account to make transactions on Ethereum blockchain network wants to use the e-commerce platform.

Steps:

1. The user logs into the MetaMask account before registering with the platform.



Blockchain E-commerce Platform

Loading...

Blockchain E-commerce Platform

User Section

MetaMask Account Address:

Role:

Unregistered

Username:

Email:

IC Number:

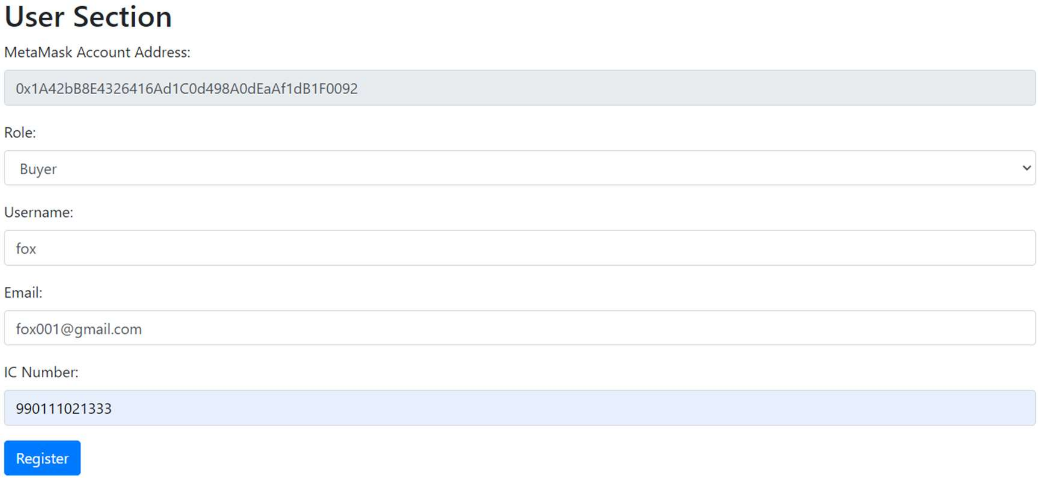
Welcome back!

The decentralized web awaits

Password

Figure 4-4-1-1 Screenshot of MetaMask UI

2. User fills up all the fields to register. User can register as buyer or seller.



User Section

MetaMask Account Address:

0x1A42bB8E4326416Ad1C0d498A0dEaAf1dB1F0092

Role:

Buyer

Username:

fox

Email:

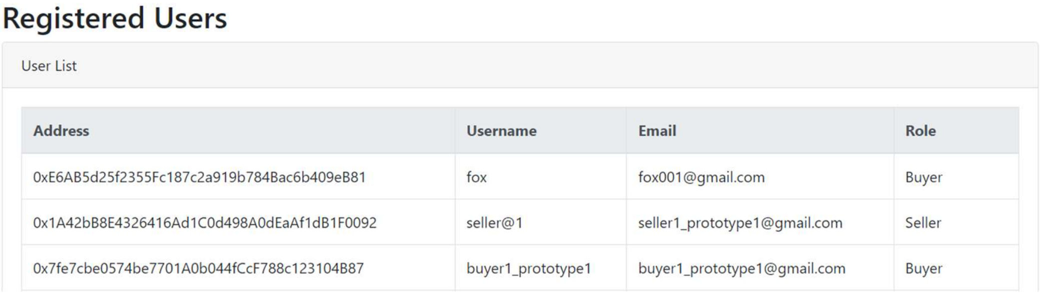
fox001@gmail.com

IC Number:

990111021333

Figure 4-4-1-2 Screenshot of User Section Test Case

3. The user will be registered and displayed (in the first row) in the User List. The IC number will be hashed before being stored and will not display in the User List.



Registered Users

User List

Address	Username	Email	Role
0xE6AB5d25f2355Fc187c2a919b784Bac6b409eB81	fox	fox001@gmail.com	Buyer
0x1A42bB8E4326416Ad1C0d498A0dEaAf1dB1F0092	seller@1	seller1_prototype1@gmail.com	Seller
0x7fe7cbe0574be7701A0b044fCcF788c123104B87	buyer1_prototype1	buyer1_prototype1@gmail.com	Buyer

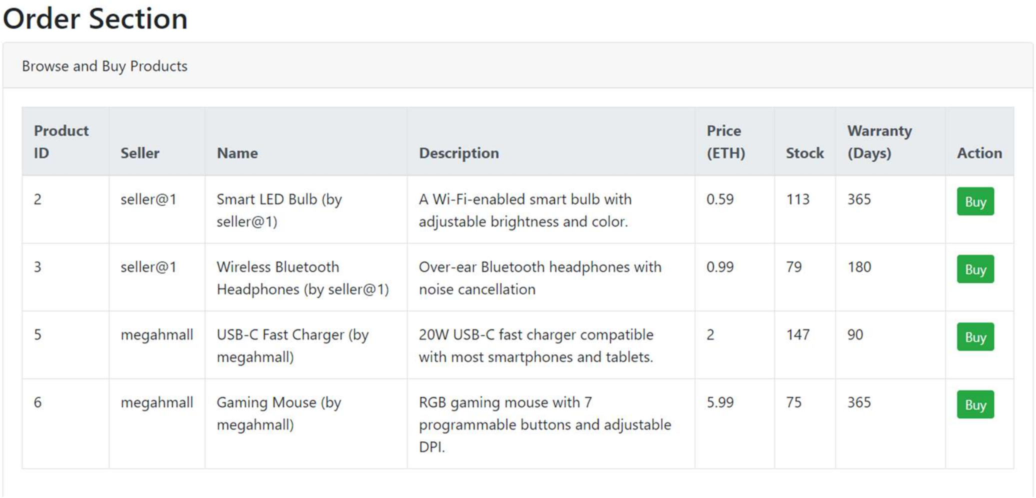
Figure 4-4-1-3 Screenshot of User List Test Case

4.4.2 Buyer Purchase Scenario

A buyer purchases a product on the platform. After the transaction is done, the stock for the product is updated. The order is updated and displayed in the Order List. The payment is processed via MetaMask on the blockchain.

Steps:

1. Buyer browses the products listed on the platform.



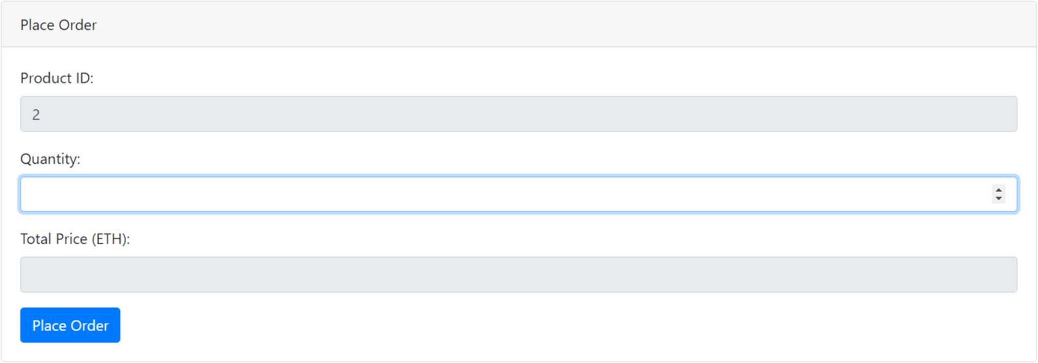
Order Section

Browse and Buy Products

Product ID	Seller	Name	Description	Price (ETH)	Stock	Warranty (Days)	Action
2	seller@1	Smart LED Bulb (by seller@1)	A Wi-Fi-enabled smart bulb with adjustable brightness and color.	0.59	113	365	Buy
3	seller@1	Wireless Bluetooth Headphones (by seller@1)	Over-ear Bluetooth headphones with noise cancellation	0.99	79	180	Buy
5	megahmall	USB-C Fast Charger (by megahmall)	20W USB-C fast charger compatible with most smartphones and tablets.	2	147	90	Buy
6	megahmall	Gaming Mouse (by megahmall)	RGB gaming mouse with 7 programmable buttons and adjustable DPI.	5.99	75	365	Buy

Figure 4-4-2-1 Screenshot of Order Section Test Case

2. Buyer selects a product and clicks "Buy" button. Buyer is directed to the "Place Order" section.



Place Order

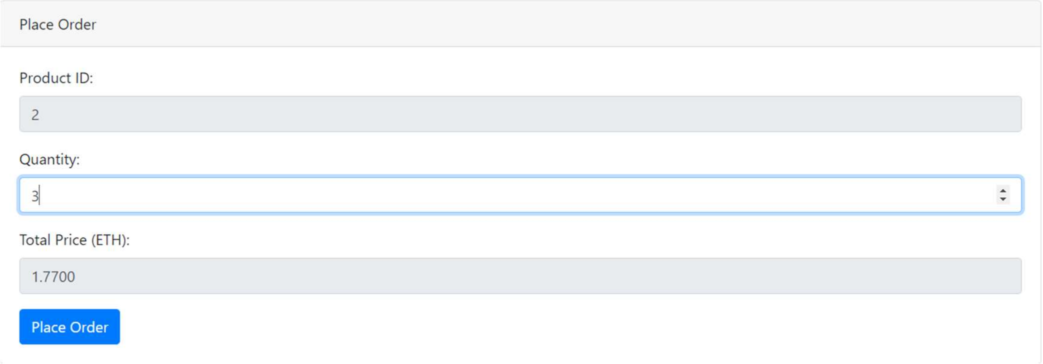
Product ID: 2

Quantity:

Total Price (ETH):

Figure 4-4-2-2 Screenshot of Place Order Section Test Case

3. Buyer enters the quantity to purchase. The Total Price will be calculated in ETH. In this case, the Total Price is *0.59ETH* x *3 = 1.77ETH.*



Place Order

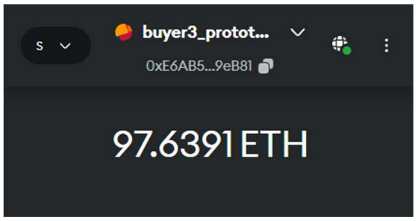
Product ID: 2

Quantity: 3

Total Price (ETH): 1.7700

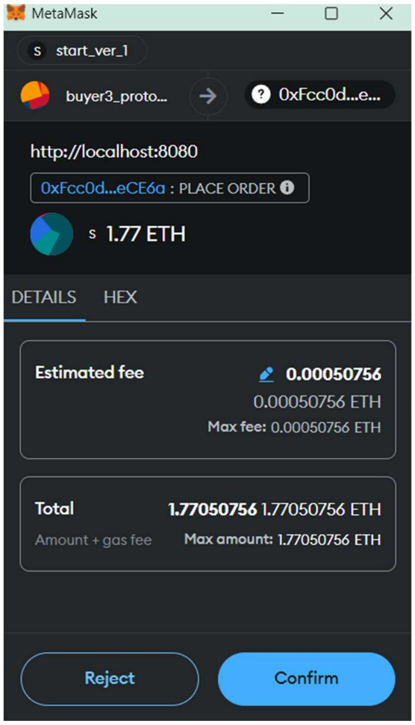
Figure 4-4-2-3 Screenshot of Total Price

4. After buyer clicked "Place Order" button, buyer needs to confirm the transaction via MetaMask to proceed the transaction. The buyer has '97.6391ETH' before the transaction.



97.6391 ETH

Figure 4-4-2-4 Screenshot of Buyer Account Before Purchase



http://localhost:8080

0xFcc0d...eCE6a : PLACE ORDER

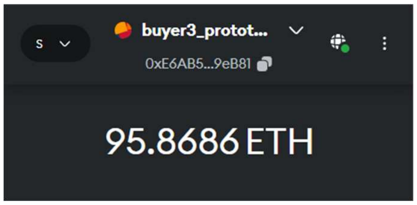
S 1.77 ETH

Estimated fee 0.00050756
0.00050756 ETH
Max fee: 0.00050756 ETH

Total 1.77050756 1.77050756 ETH
Amount + gas fee Max amount: 1.77050756 ETH

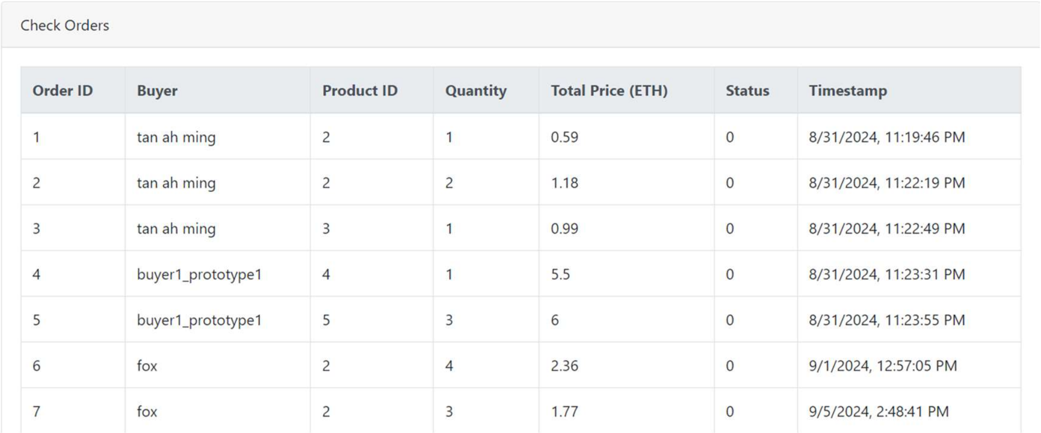
Figure 4-4-2-5 Screenshot of MetaMask Buy Transaction

5. The amount of ETH in MetaMask wallet is reflected. The order just been placed has been added to the last row of Order List with all the relevant details.



95.8686 ETH

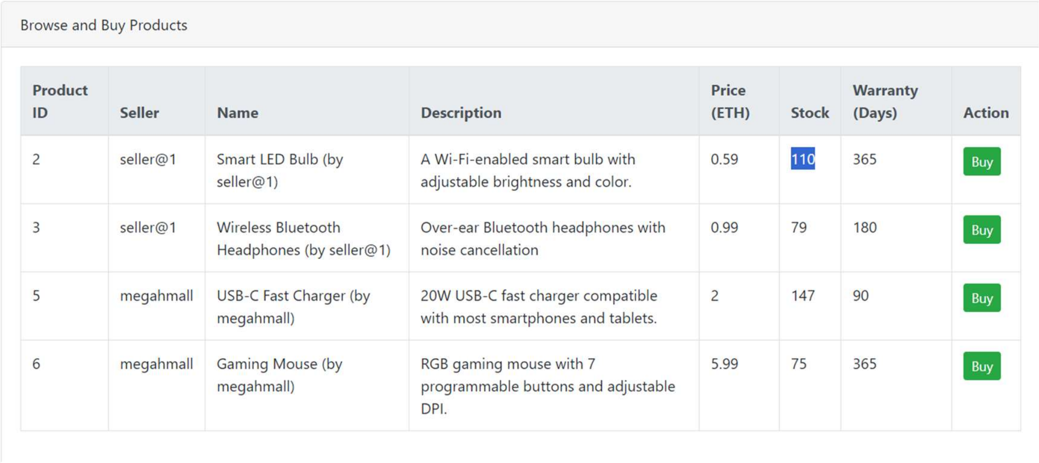
Figure 4-4-2-6 Screenshot of Buyer Account After Purchase


Check Orders

Order ID	Buyer	Product ID	Quantity	Total Price (ETH)	Status	Timestamp
1	tan ah ming	2	1	0.59	0	8/31/2024, 11:19:46 PM
2	tan ah ming	2	2	1.18	0	8/31/2024, 11:22:19 PM
3	tan ah ming	3	1	0.99	0	8/31/2024, 11:22:49 PM
4	buyer1_prototype1	4	1	5.5	0	8/31/2024, 11:23:31 PM
5	buyer1_prototype1	5	3	6	0	8/31/2024, 11:23:55 PM
6	fox	2	4	2.36	0	9/1/2024, 12:57:05 PM
7	fox	2	3	1.77	0	9/5/2024, 2:48:41 PM

Figure 4-4-2-7 Screenshot of Order List After Purchase

6. The product stock of "Smart LED Bulb" is deducted from 113 to 110, reflecting the 3 units of "Smart LED Bulb" purchased by buyer.



Browse and Buy Products

Product ID	Seller	Name	Description	Price (ETH)	Stock	Warranty (Days)	Action
2	seller@1	Smart LED Bulb (by seller@1)	A Wi-Fi-enabled smart bulb with adjustable brightness and color.	0.59	110	365	Buy
3	seller@1	Wireless Bluetooth Headphones (by seller@1)	Over-ear Bluetooth headphones with noise cancellation	0.99	79	180	Buy
5	megahmall	USB-C Fast Charger (by megahmall)	20W USB-C fast charger compatible with most smartphones and tablets.	2	147	90	Buy
6	megahmall	Gaming Mouse (by megahmall)	RGB gaming mouse with 7 programmable buttons and adjustable DPI.	5.99	75	365	Buy

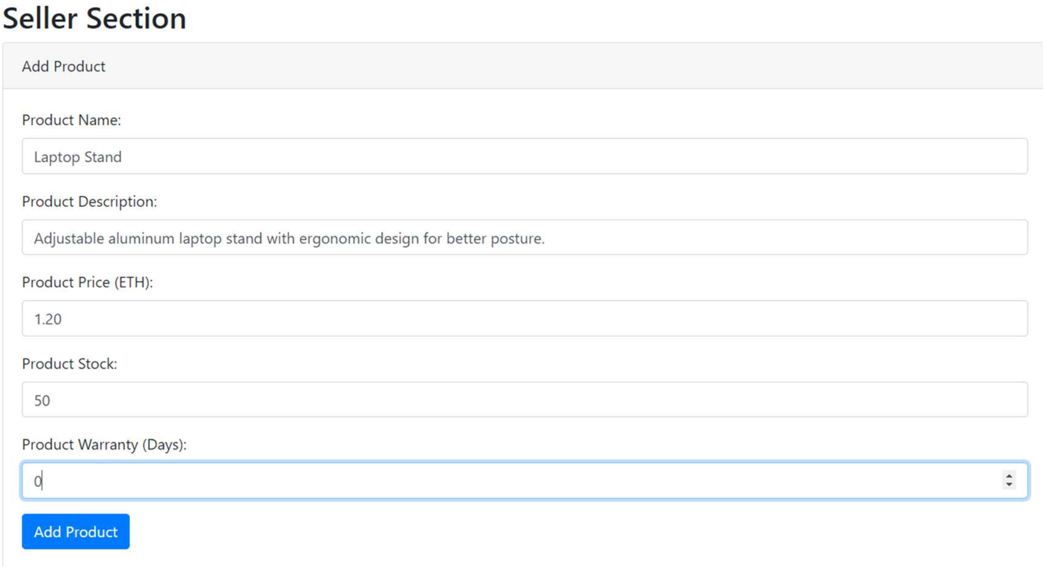
Figure 4-4-2-8 Screenshot of Product List After Purchase

4.4.3 Seller Add Product Scenario

A seller lists a new product for sale on the platform. The product is recorded on the blockchain and displayed in the listed Product List. The product is available for buyer to purchase.

Steps:

1. Seller fills up all the required product information fields to add a product to listing.



Seller Section

Add Product

Product Name:

Laptop Stand

Product Description:

Adjustable aluminum laptop stand with ergonomic design for better posture.

Product Price (ETH):

1.20

Product Stock:

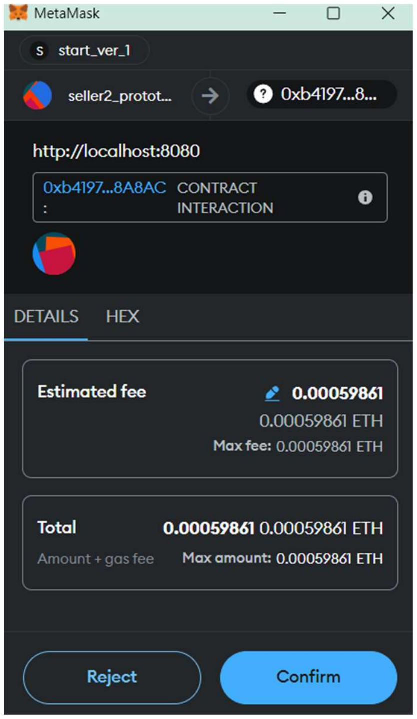
50

Product Warranty (Days):

0

Figure 4-4-3-1 Screenshot of Seller Section Adding Product

2. Seller needs to confirm the action in MetaMask. The action to add to a product to blockchain is considered as a transaction. A small of amount of gas fee is required.



http://localhost:8080

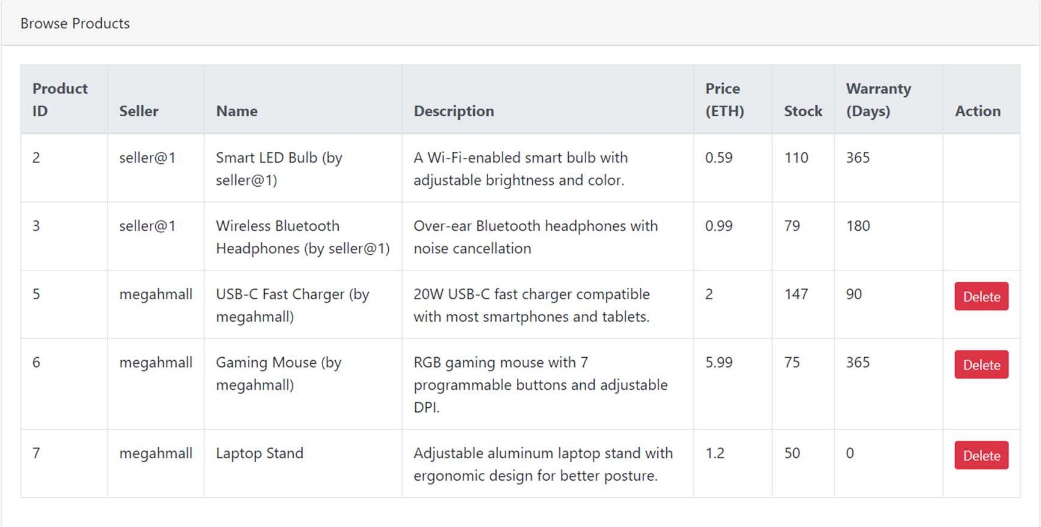
0xb4197...8A8AC : CONTRACT INTERACTION

Estimated fee 0.00059861
0.00059861 ETH
Max fee: 0.00059861 ETH

Total 0.00059861 0.00059861 ETH
Amount + gas fee Max amount: 0.00059861 ETH

Figure 4-4-3-2 Screenshot of MetaMask Add Product Transaction

3. After confirmation, the new product “Laptop Stand” is listed (in last row) in the Product List. And seller can perform actions like deleting his or her own product.



Browse Products

Product ID	Seller	Name	Description	Price (ETH)	Stock	Warranty (Days)	Action
2	seller@1	Smart LED Bulb (by seller@1)	A Wi-Fi-enabled smart bulb with adjustable brightness and color.	0.59	110	365	
3	seller@1	Wireless Bluetooth Headphones (by seller@1)	Over-ear Bluetooth headphones with noise cancellation	0.99	79	180	
5	megahmall	USB-C Fast Charger (by megahmall)	20W USB-C fast charger compatible with most smartphones and tablets.	2	147	90	Delete
6	megahmall	Gaming Mouse (by megahmall)	RGB gaming mouse with 7 programmable buttons and adjustable DPI.	5.99	75	365	Delete
7	megahmall	Laptop Stand	Adjustable aluminum laptop stand with ergonomic design for better posture.	1.2	50	0	Delete

Figure 4-4-3-3 Screenshot of Product List After Adding Product

4.5 Feasibility Analysis

The preliminary works show the feasibility of the blockchain e-commerce platform. The core functionalities could be deployed successfully and executed, showing that the platform can operate the e-commerce operations in a decentralized way.

However, some advanced functions still need to be developed and integrated to deliver a functional e-commerce platform. Furthermore, a front-end interface that is more user-friendly could be integrated to enhance the overall user experience.

CHAPTER 5

System Implementation

5.1 Hardware Setup

Hardware setup provided the environment required for testing and development of the Flutter-based e-commerce app. Development was conducted on a computer with the specifications listed in Table of Hardware Tools (Table 3-1-2-1). No physical devices were used to test, but an Android emulator provided by Android Studio was used, configured to simulate an Android 15.0 device.

5.2 Software Setup

The software tools and libraries were installed to support the development of the Flutter application and its blockchain integration. Table of Software Tools (Table A-1) lists the key software tools used.

Flutter was downloaded from the official site and installed in Visual Studio Code to enable Dart-based development. Android Studio was installed to provide the emulator for testing, checking the application's performance on a simulated Android device.

5.3 Setting and Configuration

The development environment required additional configurations for blockchain and Flutter integration. Flutter was set up by downloading the Flutter SDK from the official website and installing it with Visual Studio Code. The Dart extension was installed to support the development of Flutter. Android Studio was set up with a virtual device through the Virtual Device Manager, creating a new virtual device with the specifications listed in Table 5-3-1.

Description	**Specification**
API Level	35

Resolution (px)	1080 x 2400
Resolution (dp)	412 x 915
Density	420 dpi
ABI List	x86_64
Size on Disk	15.3 GB

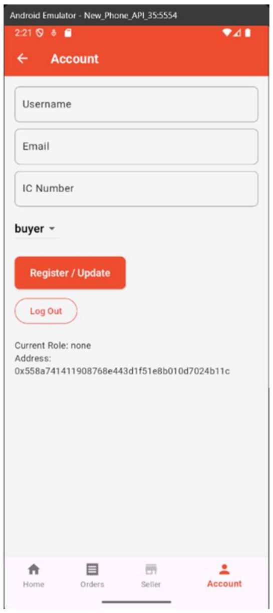
Table 5-3-1 Virtual Device Specifications

Ganache was configured to use port 7545 for blockchain integration and provided 10 test accounts for Ethereum-based smart contract deployment and testing. This enabled the simulation of blockchain transactions in the e-commerce application.

5.4 System Operation

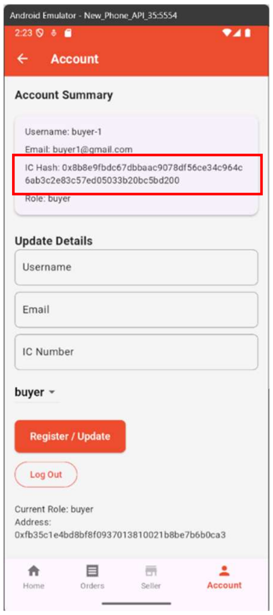
5.4.1 User Management

The user management module handled with registering users, assigning roles (buyer or seller), and hashing identity cards number (IC) for security. The users first enter the private key of a Ganache wallet. In case they are not registered, they must register as a buyer or seller, where their IC is hashed for privacy.



Current Role: none
Address:
0x558a741411908768e443d1f51e8b010d7024b11c

Figure 5-4-1-1 User Registration Screen



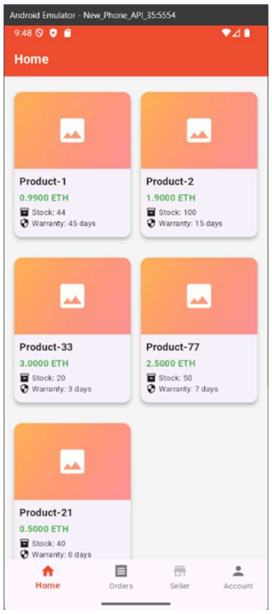
Username: buyer-1
Email: buyer1@gmail.com
IC Hash: 0x8b8e9fbdc67dbbaac9078df56ce34c964c6ab3c2e83c57ed05033b20bc5bd200
Role: buyer

Current Role: buyer
Address:
0xfb35c1e4bd8bf8f0937013810021b8be7b6b0ca3

Figure 5-4-1-2 IC Hashed

5.4.2 Product Management

The product management module allowed sellers to list products and update stock levels. The Flutter interface enabled sellers to input product details, such as name, price, and stock level.



Product-1
0.9900 ETH
Stock: 44
Warranty: 45 days

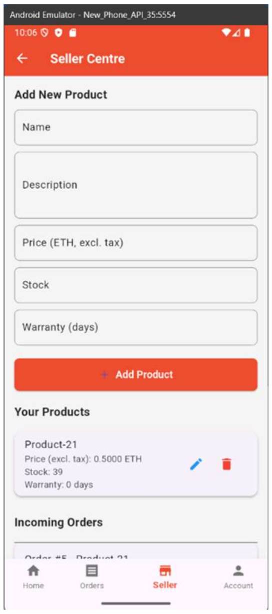
Product-2
1.9000 ETH
Stock: 100
Warranty: 15 days

Product-33
3.0000 ETH
Stock: 20
Warranty: 3 days

Product-77
2.5000 ETH
Stock: 50
Warranty: 7 days

Product-21
0.5000 ETH
Stock: 40
Warranty: 0 days

Figure 5-4-2-1 Home (Product Listing Page)



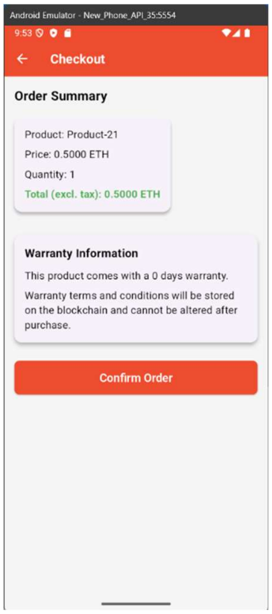
Product-21
Price (excl. tax): 0.5000 ETH
Stock: 39
Warranty: 0 days

Figure 5-4-2-2 Add Product Section

5.4.3 Order Management

The order management module supported operations for both buyers and sellers, including payment processing and status tracking. Below are the operations for each role:

Buyer Operations: Buyers browse listed products, select items to purchase, and initiate payments through MetaMask, which interacts with the Ethereum blockchain to process transactions securely. After payment, buyers can track the order status (pending, shipped, delivered, cancelled) via the Flutter interface.



Order Summary

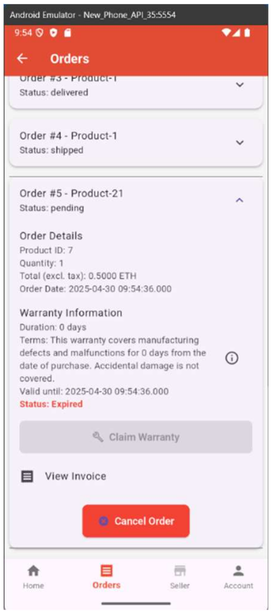
Product: Product-21
Price: 0.5000 ETH
Quantity: 1
Total (excl. tax): 0.5000 ETH

Warranty Information

This product comes with a 0 days warranty.

Warranty terms and conditions will be stored on the blockchain and cannot be altered after purchase.

Figure 5-4-3-1 Order and Payment Confirmation Screen



Order #3 - Product-1
Status: delivered

Order #4 - Product-1
Status: shipped

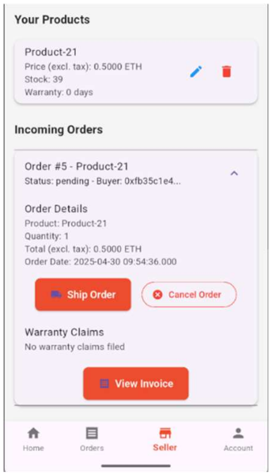
Order #5 - Product-21
Status: pending

Order Details
Product ID: 7
Quantity: 1
Total (excl. tax): 0.5000 ETH
Order Date: 2025-04-30 09:54:36.000

Warranty Information
Duration: 0 days
Terms: This warranty covers manufacturing defects and malfunctions for 0 days from the date of purchase. Accidental damage is not covered.
Valid until: 2025-04-30 09:54:36.000
Status: Expired

Figure 5-4-3-2 Order Screen

Seller Operations: Sellers manage incoming orders by reviewing buyer orders, confirming product availability, and updating order statuses. The Flutter interface provides a dashboard as known as Seller Centre where sellers can view all orders, accept or reject them, and record status changes on the blockchain. This ensures transparency, as all updates are logged immutably.



Your Products

Product-21
Price (excl. tax): 0.5000 ETH
Stock: 39
Warranty: 0 days

Incoming Orders

Order #5 - Product-21
Status: pending - Buyer: 0xfb35c1e4...

Order Details
Product: Product-21
Quantity: 1
Total (excl. tax): 0.5000 ETH
Order Date: 2025-04-30 09:54:36.000

Warranty Claims
No warranty claims filed

Figure 5-4-3-3 Order Management in Seller Center

5.5 Implementation Issues and Challenges

Several challenges were encountered during the implementation.

Initially, Web3.dart library integration with Flutter was challenging due to connectivity issues. By conducting adequate research, the issue was resolved by debugging Web3.dart configurations and ensuring proper Ethereum node connectivity.

The feature enhancements on the smart contracts' features increased the gas required for the execution of smart contracts, making some of the functions failed without notification. After inspection, it was because the gas required excessed the gas limit set. Hence, the gas limit was increased without affecting the overall performance of the system, and the code was optimized by eliminating redundant operations to reduce gas consumption.

The transition from the web page prototype to the mobile app caused UI issues due to unfamiliarity with Flutter development. These issues were resolved through iterative debugging and testing. Knowledge learned from previous course taken was also applied to ensure mobile friendliness and good user experience.

5.6 Concluding Remark

The Flutter based e-commerce application was successfully implemented with integrating Ethereum blockchain functionality via Ganache and MetaMask. The system supported user management with IC hashing enhancing security, product listing, and order processing. The next chapter evaluates the system's performance and usability against the project objectives.

CHAPTER 6

System Evaluation and Discussion

6.1 Evaluation Methodology

The system was tested to evaluate the performance against the three primary objectives. The testing methods and metrics were designed to measure the ability of the system to reduce intermediary costs, provide a user-friendly user interface, and have transparent transaction records.

6.1.1 Use Case Testing

To verify the system's functional capabilities to eliminate intermediaries and provide transparent transactions, comprehensive use case testing was performed. This involved testing end-to-end user flows through the system for sellers and buyers, documenting visually every step to confirm functionality.

Key use cases tested included:

- User registration and authentication
- Product listing and management by sellers
- Product browsing and searching by buyers
- Order placement and payment processing
- Order status tracking
- Transaction record verification and invoice record on the blockchain

6.1.2 Performance Metrics

To quantify the system's effectiveness, the following performance metrics were established:

Intermediary Fee Metrics:

- Gas fees for smart contract execution. This is to be compared with traditional platform fees.

Transparency Metrics:

- Accuracy of blockchain-logged transaction records.
- Traceability of product and order history.
- Immutability of generated invoices and transaction logs.

User Experience Metrics:

- Transaction completion time.
- User satisfaction scores from usability testing.

6.1.3 Security Assessment

The security assessment focused on checking the integrity of data with privacy and transparency within the system. This included the verification of IC hashing to protect user identity, and the immutability of blockchain records.

6.2 Testing Environment

The testing was conducted in an environment with typical user conditions while providing controlled parameters for evaluation:

- Hardware/Software Setup: Android emulator (API Level 35, 1080x2400 resolution).
- Blockchain Environment: Ganache testnet configured on port 7545 with 10 test accounts.
- Testing Period: Two weeks of continuous testing with simulated user scenarios.
- Test Users: A combination of developer and volunteer testers.

6.3 Use Case Testing Results

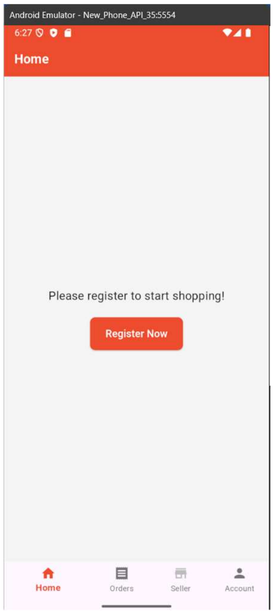
The use case testing demonstrated the system's ability to support end-to-end e-commerce operations without traditional intermediaries.

6.3.1 Seller Use Case

The seller journey testing verified that sellers could effectively:

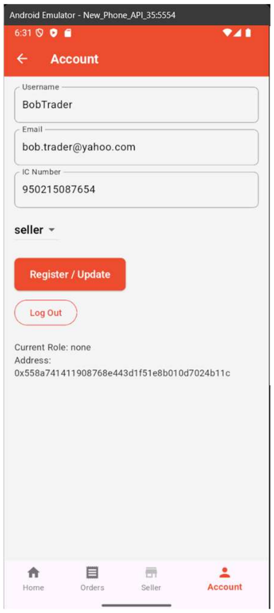
- Create and manage product listings
- Receive order notifications
- Access transaction records and sales history
- Generate blockchain-verified invoices

The seller dashboard successfully displayed all relevant information for order management and transaction tracking.



Please register to start shopping!

Figure 6-3-1-1 Home Page



Android Emulator - New_Phone_API_35:5554

6:31

Account

Username: BobTrader

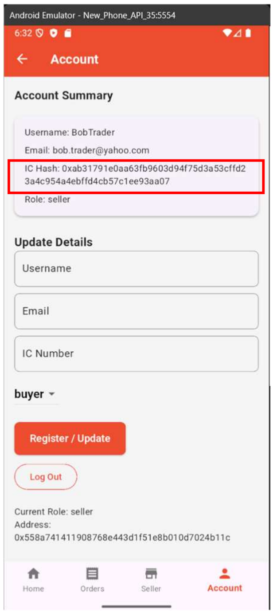
Email: bob.trader@yahoo.com

IC Number: 950215087654

seller

Current Role: none
Address:
0x558a741411908768e443d1f51e8b010d7024b11c

Figure 6-3-1-2 Seller Registration



Android Emulator - New_Phone_API_35:5554

6:32

Account

Account Summary

Username: BobTrader
Email: bob.trader@yahoo.com
IC Hash: 0xab31791e0aa63fb9603d94f75d3a53cffd23a4c954a4ebffd4cb57c1ee93aa07
Role: seller

Update Details

Username

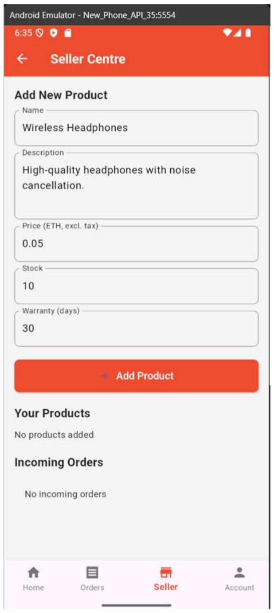
Email

IC Number

buyer

Current Role: seller
Address:
0x558a741411908768e443d1f51e8b010d7024b11c

Figure 6-3-1-3 Hashed IC Storage on Blockchain



Add New Product

Name: Wireless Headphones

Description: High-quality headphones with noise cancellation.

Price (ETH, excl. tax): 0.05

Stock: 10

Warranty (days): 30

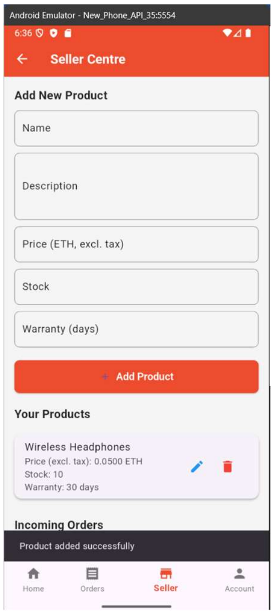
Your Products

No products added

Incoming Orders

No incoming orders

Figure 6-3-1-4 Adding New Product



Add New Product

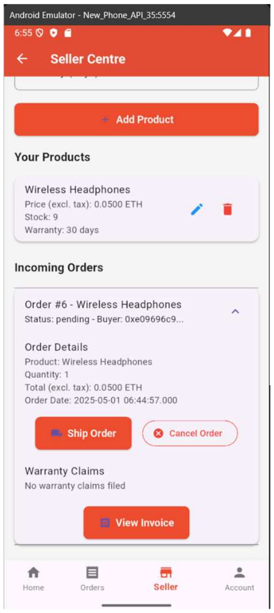
Your Products

Wireless Headphones
Price (excl. tax): 0.0500 ETH
Stock: 10
Warranty: 30 days

Incoming Orders

Product added successfully

Figure 6-3-1-5 Product Added Successfully



Your Products

Wireless Headphones
Price (excl. tax): 0.0500 ETH
Stock: 9
Warranty: 30 days

Incoming Orders

Order #6 - Wireless Headphones
Status: pending - Buyer: 0xe09696c9...

Order Details
Product: Wireless Headphones
Quantity: 1
Total (excl. tax): 0.0500 ETH
Order Date: 2025-05-01 06:44:57.000

Warranty Claims
No warranty claims filed

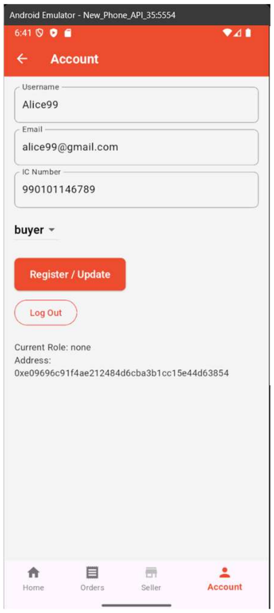
Figure 6-3-1-6 Incoming Orders in Seller Center

6.3.2 Buyer Use Case

The buyer journey testing confirmed that buyers could successfully:

- Register with the system using secure IC hashing
- Browse products with intuitive navigation
- Purchase a product
- Complete checkout
- View order history and transaction records

All buyer actions were successfully recorded on the blockchain, with transaction hashes available for verification.



Android Emulator - New_Phone_API_35:5554

Account

Username: Alice99

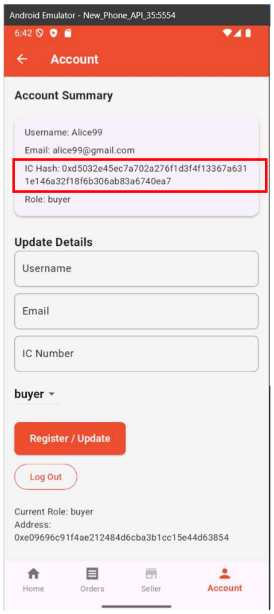
Email: alice99@gmail.com

IC Number: 990101146789

buyer

Current Role: none
Address:
0xe09696c91f4ae212484d6cba3b1cc15e44d63854

Figure 6-3-2-1 Buyer Registration



Android Emulator - New_Phone_API_35:5554

Account

Account Summary

Username: Alice99
Email: alice99@gmail.com
IC Hash: 0xd5032e45ec7a702a276f1d3f4f13367a631
1e146a32f18f6b306ab83a6740ea7
Role: buyer

Update Details

Username

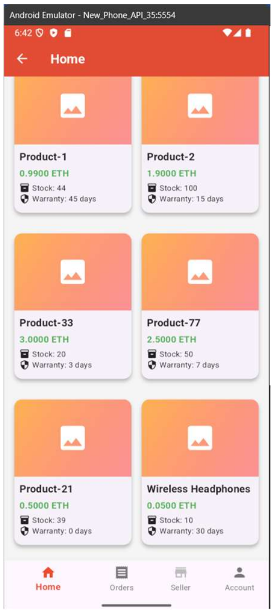
Email

IC Number

buyer

Current Role: buyer
Address:
0xe09696c91f4ae212484d6cba3b1cc15e44d63854

Figure 6-3-2-2 Hashed IC Storage for Buyer



Product-1
0.9900 ETH
Stock: 44
Warranty: 45 days

Product-2
1.9000 ETH
Stock: 100
Warranty: 15 days

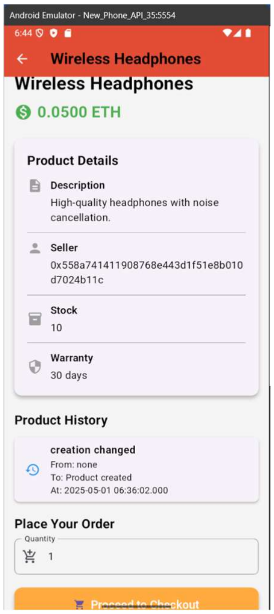
Product-33
3.0000 ETH
Stock: 20
Warranty: 3 days

Product-77
2.5000 ETH
Stock: 50
Warranty: 7 days

Product-21
0.5000 ETH
Stock: 39
Warranty: 0 days

Wireless Headphones
0.0500 ETH
Stock: 10
Warranty: 30 days

Figure 6-3-2-3 Browse Products



Wireless Headphones
0.0500 ETH

Product Details

Description
High-quality headphones with noise cancellation.

Seller
0x558a741411908768e443d1f51e8b010d7024b11c

Stock
10

Warranty
30 days

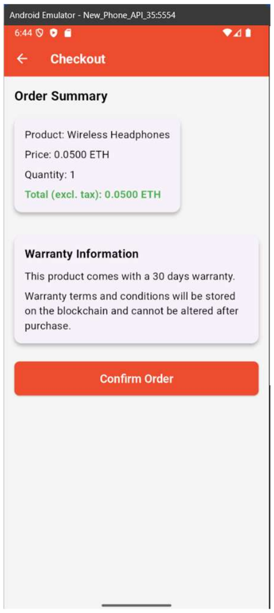
Product History

creation changed
From: none
To: Product created
At: 2025-05-01 06:36:02.000

Place Your Order

Quantity
1

Figure 6-3-2-4 Product Details



Android Emulator - New_Phone_API_35:5554

6:44

Order Summary

Product: Wireless Headphones

Price: 0.0500 ETH

Quantity: 1

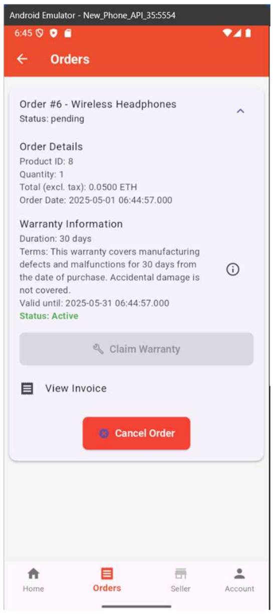
Total (excl. tax): 0.0500 ETH

Warranty Information

This product comes with a 30 days warranty.

Warranty terms and conditions will be stored on the blockchain and cannot be altered after purchase.

Figure 6-3-2-5 Order Checkout Screen



Android Emulator - New_Phone_API_35:5554

6:45

Order #6 - Wireless Headphones
Status: pending

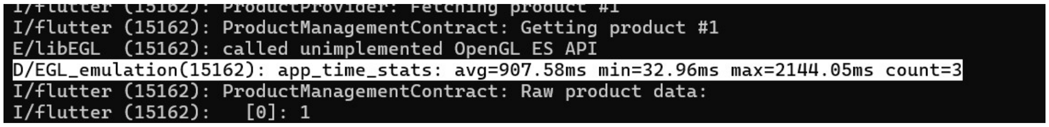
Order Details
Product ID: 8
Quantity: 1
Total (excl. tax): 0.0500 ETH
Order Date: 2025-05-01 06:44:57.000

Warranty Information
Duration: 30 days
Terms: This warranty covers manufacturing defects and malfunctions for 30 days from the date of purchase. Accidental damage is not covered.
Valid until: 2025-05-31 06:44:57.000
Status: Active

Figure 6-3-2-6 View Orders

6.4 Performance Results and Analysis

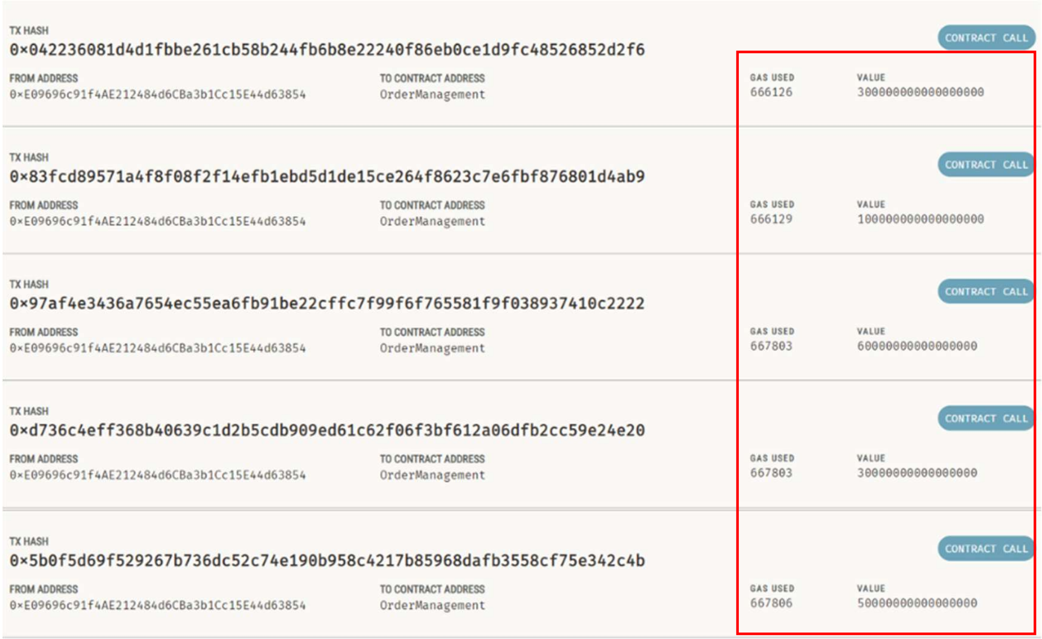
6.4.1 Transaction Processing Performance



```
I/flutter (15162): ProductProvider: Fetching product #1
I/flutter (15162): ProductManagementContract: Getting product #1
E/libEGL  (15162): called unimplemented OpenGL ES API
D/EGL_emulation(15162): app_time_stats: avg=907.58ms min=32.96ms max=2144.05ms count=3
I/flutter (15162): ProductManagementContract: Raw product data:
I/flutter (15162):   [0]: 1
```

Figure 6-4-1-1 Transaction Processing Time Log

All the transactions are successful on the system. The average transaction time is 907.58ms, which is less than one second.



TX HASH
0×042236081d4d1fbbe261cb58b244fb6b8e22240f86eb0ce1d9fc48526852d2f6
FROM ADDRESS 0×E09696c91f4AE212484d6CBa3b1Cc15E44d63854
TO CONTRACT ADDRESS OrderManagement
GAS USED 666126
VALUE 300000000000000000

TX HASH
0×83fcd89571a4f8f08f2f14efb1ebd5d1de15ce264f8623c7e6fbf876801d4ab9
FROM ADDRESS 0×E09696c91f4AE212484d6CBa3b1Cc15E44d63854
TO CONTRACT ADDRESS OrderManagement
GAS USED 666129
VALUE 100000000000000000

TX HASH
0×97af4e3436a7654ec55ea6fb91be22cffc7f99f6f765581f9f038937410c2222
FROM ADDRESS 0×E09696c91f4AE212484d6CBa3b1Cc15E44d63854
TO CONTRACT ADDRESS OrderManagement
GAS USED 667803
VALUE 60000000000000000

TX HASH
0×d736c4eff368b40639c1d2b5cdb909ed61c62f06f3bf612a06dfb2cc59e24e20
FROM ADDRESS 0×E09696c91f4AE212484d6CBa3b1Cc15E44d63854
TO CONTRACT ADDRESS OrderManagement
GAS USED 667803
VALUE 30000000000000000

TX HASH
0×5b0f5d69f529267b736dc52c74e190b958c4217b85968dafb3558cf75e342c4b
FROM ADDRESS 0×E09696c91f4AE212484d6CBa3b1Cc15E44d63854
TO CONTRACT ADDRESS OrderManagement
GAS USED 667806
VALUE 50000000000000000

Figure 6-4-1-2 Ganache Transaction Log for Order Placement

Transaction	**Gas Used (Transaction Fee) in Wei**	**Value (Product Value) in Wei**	**Fee/Value Ratio**
1	666,126	3E+17	2.22E-12
2	666,129	1E+17	6.66E-12
3	667,803	6E+16	1.11E-11
4	667,803	3E+16	2.23E-11
5	667,806	5E+16	1.34E-11
Average	667,133	1.08E+17	1.11E-11

Table 6-4-1-1 Gas Fee and Value Per Transaction

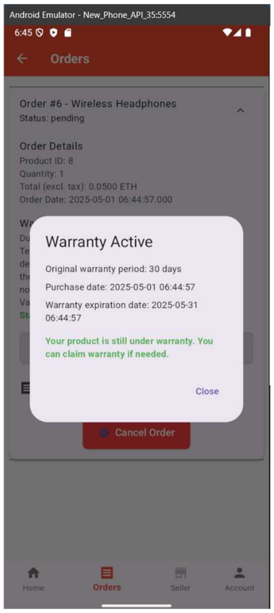
From the table, the average gas fee for an order transaction is 667,133 Wei, which converted to approximately 0.000000000000667133 ETH. At the rate of $1,806.36 per Ether, this equals approximately $0 in USD. Compared to Shopee's transaction fee of 3.78% [15], which meant the transaction fee will scale with the order amount, our system provided advantages. This performance is good for e-commerce applications but will require optimization for mainnet deployment.

6.4.2 Security and Transparency Verification

The system showed good performance in maintaining transparent yet secure records.

IC hashing correctly protected user identity while maintaining accountability.

Transaction records and invoices were successfully stored on the blockchain and remained immutable.



Order #6 - Wireless Headphones
Status: pending

Order Details
Product ID: 8
Quantity: 1
Total (excl. tax): 0.0500 ETH
Order Date: 2025-05-01 06:44:57.000

Warranty Active

Original warranty period: 30 days

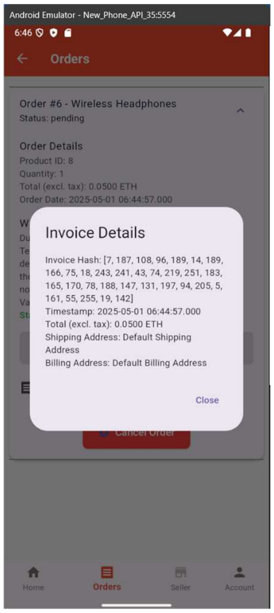
Purchase date: 2025-05-01 06:44:57

Warranty expiration date: 2025-05-31 06:44:57

Your product is still under warranty. You can claim warranty if needed.

Close

Figure 6-4-2-1 Product Warranty Information



Order #6 - Wireless Headphones
Status: pending

Order Details
Product ID: 8
Quantity: 1
Total (excl. tax): 0.0500 ETH
Order Date: 2025-05-01 06:44:57.000

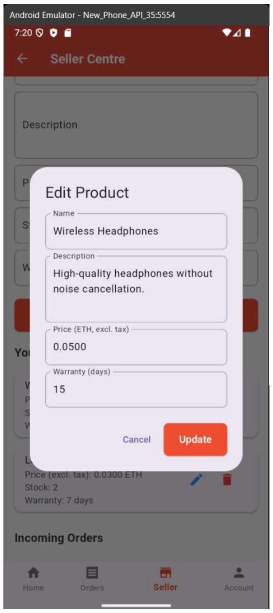
Invoice Details

Invoice Hash: [7, 187, 108, 96, 189, 14, 189, 166, 75, 18, 243, 241, 43, 74, 219, 251, 183, 165, 170, 78, 188, 147, 131, 197, 94, 205, 5, 161, 55, 255, 19, 142]
Timestamp: 2025-05-01 06:44:57.000
Total (excl. tax): 0.0500 ETH
Shipping Address: Default Shipping Address
Billing Address: Default Billing Address

Close

Figure 6-4-2-2 Invoice Details

Product updates were traceable through the blockchain history.



Android Emulator - New_Phone_API_35:5554

7:20

Seller Centre

Description

Edit Product

Name: Wireless Headphones

Description: High-quality headphones without noise cancellation.

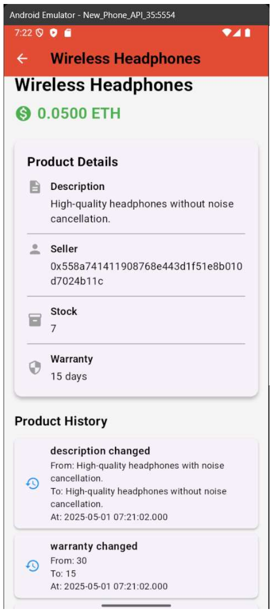
Price (ETH, excl. tax): 0.0500

Warranty (days): 15

Price (excl. tax): 0.0300 ETH
Stock: 2
Warranty: 7 days

Incoming Orders

Figure 6-4-2-3 Edit Product Details



Android Emulator - New_Phone_API_35:5554

7:22

Wireless Headphones

Wireless Headphones

0.0500 ETH

Product Details

Description
High-quality headphones without noise cancellation.

Seller
0x558a741411908768e443d1f51e8b010d7024b11c

Stock
7

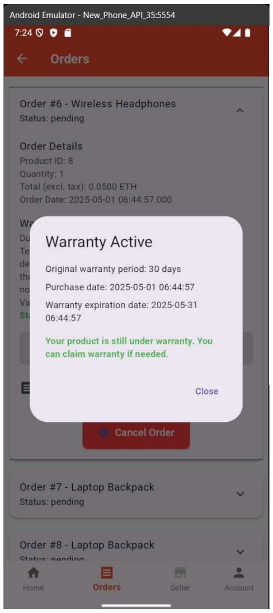
Warranty
15 days

Product History

description changed
From: High-quality headphones with noise cancellation.
To: High-quality headphones without noise cancellation.
At: 2025-05-01 07:21:02.000

warranty changed
From: 30
To: 15
At: 2025-05-01 07:21:02.000

Figure 6-4-2-4 Product Details After Edit



Android Emulator - New_Phone_API_35:5554

7:24

Orders

Order #6 - Wireless Headphones
Status: pending

Order Details
Product ID: 8
Quantity: 1
Total (excl. tax): 0.0500 ETH
Order Date: 2025-05-01 06:44:57.000

Warranty Active

Original warranty period: 30 days

Purchase date: 2025-05-01 06:44:57

Warranty expiration date: 2025-05-31 06:44:57

Your product is still under warranty. You can claim warranty if needed.

Close

Cancel Order

Order #7 - Laptop Backpack
Status: pending

Order #8 - Laptop Backpack
Status: pending

Figure 6-4-2-5 Unchanged Warranty Details Post-Edit

The warranty details especially the warranty period remained unchanged even after edits. This provided protection for buyers in case of warranty disputes, preventing sellers from modifying product descriptions to deceive customers.

6.4.3 Usability Testing Survey

A usability survey was conducted via Google Form from May 1 to May 4, 2025, to assess the user-friendliness of the application, without collecting personal information to preserve anonymity. Twenty participants, primarily aged between 18–24 with experience in e-commerce and various blockchain expertise (8 experienced, 12 inexperienced), carried out activities such as registration, product browsing, ordering, and product management, as specified in the demo scenario. A 5-point Likert scale was used in the survey (1 = Very Difficult/Poor, 5 = Very Easy/Excellent) with 12 questions, and qualitative comments for satisfaction evaluation to ensure that the usability of the project achieves the objectives.

User satisfaction for functionalities offered by e-commerce is backed up with very good survey results showing mean values 4.5 for registration, checkout at 4.5, order tracking at 4.4, and visual design also at 4.5, showing an easily understandable interface like similar well-known systems. Seller tasks such as listing products (4.6) and order management (4.6) also did well, indicating the successful organization of the Seller Center. However. blockchain tasks were less highly rated, with wallet integration (3.6) and understanding blockchain benefits (3.6) pointing to difficulties for new cryptocurrency users, who reported confusion over MetaMask and blockchain technology in comments. For example, one user responded with "More blockchain info please.".

These findings confirmed the app's success in acquiring user-friendliness to carry out core e-commerce activities with the support of high preference against traditional systems (4.1) and recommendation likelihood (4.15). The lower blockchain scores reflected the need for further user-centricity such as in-app guidance. This consideration supported the usability of the app but identified blockchain training as a key area to work on to guarantee consistency with the project objective of making an accessible blockchain-based e-commerce app.

6.5 Evaluation of Objectives Achievement

6.5.1 Reduce Transaction and Processing Fees

The system successfully reduced the intermediary transaction and processing fees. The system demonstrated significant cost advantages with an average gas fee of 667,133 Wei per order transaction. When converted to ETH (0.000000000000667133) and then to USD at the exchange rate of $1,806.36 per Ether, the transaction cost is approximately $0. This represented a significant improvement over traditional platforms like Shopee, which charges a percentage-based fee of 3.78% that scales with order value.

6.5.2 User-Friendliness

The system is designed to align with the Shopee’s interface. This will lower the resistance for users to switch to a new ecommerce platform from a giant like Shopee. Usability testing results from test users also indicated positive reception of the interface design.

6.5.3 Decentralized Transparency

The system effectively achieved decentralized transparency through immutable blockchain records of transactions and invoices, traceable product and order histories, user identification privacy.

6.6 Evaluation Challenges

Several challenges were encountered during the evaluation process. The Ganache testnet could not fully represent the real world mainnet conditions. Simulated transactions are designed as realistic as possible to approximate real-world performance.

Additionally, due to the time constraints, the usability testing was limited to a small number of users. This could potentially affect the result reliability.

However, these challenges were mitigated by focusing on controlled test scenarios and prioritizing key functionalities.

6.7 Concluding Remark

The evaluation suggested the system has fulfilled the project objectives. Preliminary results showed successful blockchain integration and user-friendly design. Conclusions and future enhancements are discussed in the following chapter.

CHAPTER 7

Conclusion and Recommendation

7.1 Conclusion

This project had managed to transform the web-based prototype into a functional, operable Flutter based mobile application, achieving the three main objectives outlined at the project's introduction: reduction of intermediary costs, usability, and decentralized transparency.

The system integrated Ethereum blockchain technology and modern mobile application development approaches to achieve a decentralized e-commerce marketplace that addressed the limitations of traditional centralized marketplaces. Through careful implementation and testing, several significant achievements have been made in both development phases.

The practical significance of this project extended beyond academic development. By eliminating intermediaries, the system showed potential for significant reduction in transaction fees compared to traditional e-commerce platforms that typically charge percentage-based fee per transaction. The transparent, immutable transaction records addressed trust issues in online commerce and reduced the need for third-party verification or resolution dispute.

Although achievements had been made, the project encountered several challenges during the implementation and evaluation. The system performance could be limited to the Ethereum network scalability under high load. The gas fee volatility might impact transaction costs for purchases that had small value. Testing limitation in the Ganache testnet might not fully represent the real world mainnet conditions.

Nevertheless, this project introduced a significant advancement in applying blockchain technology to e-commerce system and applications. The system successfully minimized the gap between technical blockchain capabilities and user needs. It created a secure, user-friendly e-commerce platform that reduces transaction fees through decentralization. The evaluation results suggested that the system aligned well with the project's primary objectives.

7.2 Recommendation

Based on the development experience and evaluation results, several recommendations are proposed for future enhancements of the blockchain based e-commerce system.

Technologically, the future development should focus on overcoming current limitations of the system by implementing several specific enhancements. Adding Ethereum Layer 2 solutions such as Polygon or Optimism would greatly reduce gas prices and increase transaction speed, making low-value transactions economically viable [16]. Making feature upgrades to smart contracts would make system improve without impacting existing users. Additionally, minimizing blockchain activity to save battery and processing power consumption would make the application more efficient for users with lower-end devices.

The user experience can be enhanced with certain adjustments to have greater accessibility and adoption. Creating more intuitive wallet integration with tutorial setup workflows and simplified approval flows for transactions would reduce technical resistance, particularly for users who lack a background in blockchain technology. Decentralized ratings, review, and verification mechanisms would enhance trust between buyer and seller without requiring centralized management, while the accommodation of a broader range of cryptocurrency payment sources would enhance flexibility. Developing interactive tutorials and user-centric help features would guide beginner users through their first fresh experiences with the platform. Incorporating advanced filtering and categorization features would facilitate product browsing while maintaining privacy and decentralization principles.

By putting these recommendations into practice, the system would be able to overcome current limitations while enhancing usability and functionality. Foundations set in this project are a good foundation upon which these enhancements can be built, probably changing how e-commerce transactions are processed through making them more effective, transparent, and fair for all parties involved.

REFERENCES

[1] K. Snyder, “35 eCommerce Statistics of 2023,” *Forbes*, Mar. 28, 2024. https://www.forbes.com/advisor/business/ecommerce-statistics/

[2] A. Cheong Pui Yin, “New E-Commerce Platform, Temu, Launches In Malaysia,” *RinggitPlus*, Sep. 11, 2023. https://ringgitplus.com/en/blog/online-shopping/new-e-commerce-platform-temu-launches-in-malaysia.html

[3] D. A. Sharma, D. S. K. Mishra, and M. V. K. Srivastav, “The Evolution And Impact Of E-Commerce,” *Journal of Namibian Studies : History Politics Culture*, vol. 33, pp. 1838–1846, May 2023, doi: https://doi.org/10.59670/jns.v33i.3260.

[4] V. Santos, T. Augusto, J. Vieira, L. Bacalhau, B. Sousa, and D. Pontes, “E-Commerce: Issues, Opportunities, Challenges, and Trends,” *E-Commerce: Issues, Opportunities, Challenges, and Trends*, Sep. 2022, doi: https://doi.org/10.4018/978-1-6684-5523-4.ch012.

[5] Z. Mustafa, M. S. A. Hassan, and M. D. A. Aziz, “(PDF) E-Commerce Challenges and Solutions,” *ResearchGate*, 2011. https://www.researchgate.net/publication/304621797_E-Commerce_Challenges_and_Solutions

[6] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,” *2017 IEEE International Congress on Big Data (BigData Congress)*, Jun. 2017, doi: https://doi.org/10.1109/BigDataCongress.2017.85.

[7] H. Treiblmaier and C. Sillaber, “The impact of blockchain on e-commerce: a framework for salient research topics,” *Electronic Commerce Research and Applications*, vol. 48, p. 101054, Apr. 2021, doi: https://doi.org/10.1016/j.elerap.2021.101054.

[8] H. Taherdoost and M. Madanchian, “Blockchain-Based E-Commerce: A Review on Applications and Challenges,” *Electronics*, vol. 12, no. 8, p. 1889, Jan. 2023, doi: https://doi.org/10.3390/electronics12081889.

[9] A. Zhang, L. Zhang, and W. Zhu, “Safety and Security of E-commerce Transactions Based on Blockchain Technology - ProQuest,” *Proquest.com*,

2024. https://www.proquest.com/openview/92f8689605326f670266e1931e85e468/1?pq-origsite=gscholar&cbl=4433095 (accessed Sep. 02, 2024).

[10] P. Rao, S. Balasubramanian, N. Vihari, S. Jabeen, V. Shukla, and J. Chanchaichujit, "The e-commerce supply chain and environmental sustainability: An empirical investigation on the online retail sector," *Cogent Business & Management*, vol. 8, no. 1, p. 1938377, Jan. 2021, doi: https://doi.org/10.1080/23311975.2021.1938377.

[11] IEA, "Data centres & networks," *IEA*, Jul. 11, 2023. https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks

[12] C. Mulligan, S. Morsfield, and E. Cheikosman, "Blockchain for sustainability: A systematic literature review for policy impact," *Telecommunications Policy*, vol. 48, no. 2, p. 102676, Oct. 2023, doi: https://doi.org/10.1016/j.telpol.2023.102676.

[13] S. Almekhlafi and N. Al-Shaibany, "The Literature Review of Blockchain Adoption," *Asian Journal of Research in Computer Science*, pp. 29–50, Mar. 2021, doi: https://doi.org/10.9734/ajrcos/2021/v7i230177.

[14] A. Esfahbodi, G. Pang, and L. Peng, "Determinants of consumers' adoption intention for blockchain technology in E-commerce," *Journal of Digital Economy*, vol. 1, no. 2, pp. 89–101, Sep. 2022, doi: https://doi.org/10.1016/j.jdec.2022.11.001.

[15] "Seller Education Hub," *seller.shopee.com.my*, Jul. 17, 2024. https://seller.shopee.com.my/edu/article/16038

[16] D. Lawrence, "Polygon vs Optimism: How Do They Provide Enhancements to Ethereum? | Cryptopolitan," *Cryptopolitan*, Dec. 19, 2024. https://www.cryptopolitan.com/polygon-vs-optimism-enhancements-to-ethereum/ (accessed May 02, 2025).

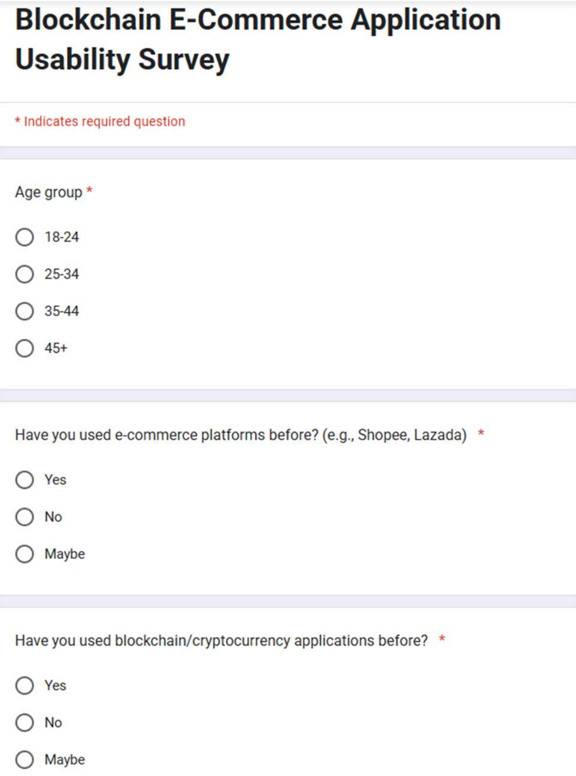
APPENDIX

A.1 Table of Software Tools

Tool	Description
Ethereum Blockchain Platform	The platform shall be developed on the Ethereum blockchain because it provides one of the robust and widely used environments for deploying smart contracts.
Solidity	Solidity is a key programming language used during the development of smart contracts on the Ethereum platform.
Truffle	Truffle is a complete development environment for Ethereum to create, test, and deploy smart contracts.
Ganache	For testing smart contracts on a local blockchain before pushing onto a public testnet or mainnet.
Flutter	Flutter is a UI software development kit that enables developers to create natively compiled applications for mobile, web, and desktop
Web3.js	Web3.js is a JavaScript library that enables interaction with the Ethereum blockchain.
GitHub	Used for version control in the project.
MetaMask	MetaMask is a browser extension, which also works as a cryptocurrency wallet interacting with Ethereum blockchain.
Visual Studio Code	Visual Studio Code is an extensible code editor that supports a wide range of programming languages, such as Solidity and JavaScript.
Android Studio	Provided the phone emulator for testing the Flutter mobile application.

Table A-1 Table of Software Tools

A.2 Usability Survey Questions



Blockchain E-Commerce Application Usability Survey

* Indicates required question

Age group *

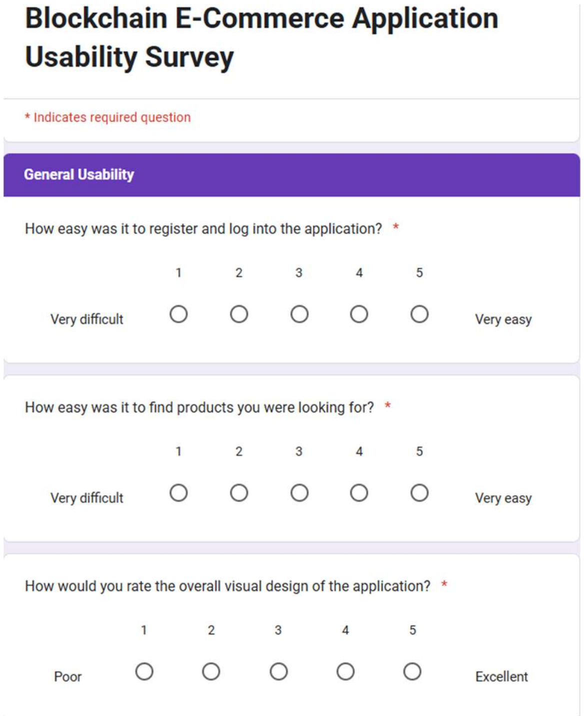
○ 18-24
○ 25-34
○ 35-44
○ 45+

Have you used e-commerce platforms before? (e.g., Shopee, Lazada) *

○ Yes
○ No
○ Maybe

Have you used blockchain/cryptocurrency applications before? *

○ Yes
○ No
○ Maybe



Blockchain E-Commerce Application Usability Survey

* Indicates required question

General Usability

How easy was it to register and log into the application? *

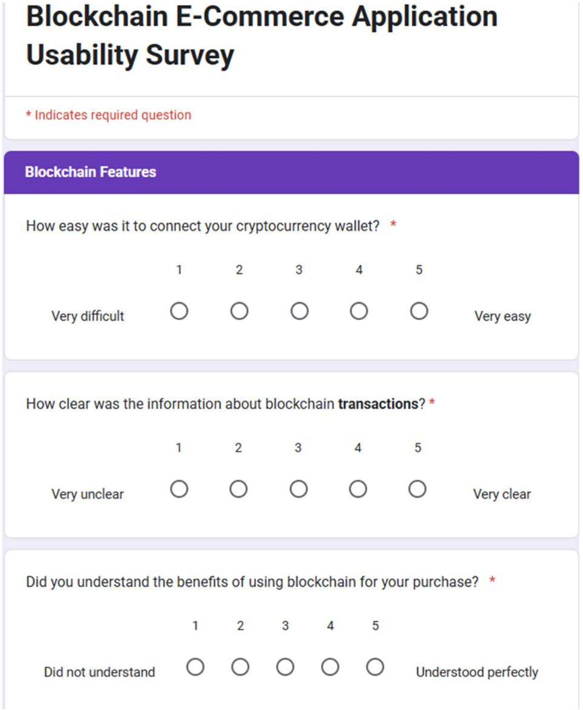
	1	2	3	4	5	
Very difficult	○	○	○	○	○	Very easy

How easy was it to find products you were looking for? *

	1	2	3	4	5	
Very difficult	○	○	○	○	○	Very easy

How would you rate the overall visual design of the application? *

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent



Blockchain E-Commerce Application Usability Survey

* Indicates required question

Blockchain Features

How easy was it to connect your cryptocurrency wallet? *

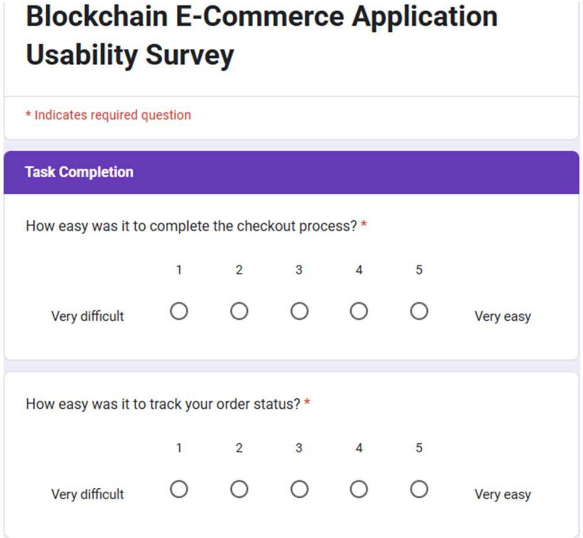
	1	2	3	4	5	
Very difficult	○	○	○	○	○	Very easy

How clear was the information about blockchain **transactions**? *

	1	2	3	4	5	
Very unclear	○	○	○	○	○	Very clear

Did you understand the benefits of using blockchain for your purchase? *

	1	2	3	4	5	
Did not understand	○	○	○	○	○	Understood perfectly



Blockchain E-Commerce Application Usability Survey

* Indicates required question

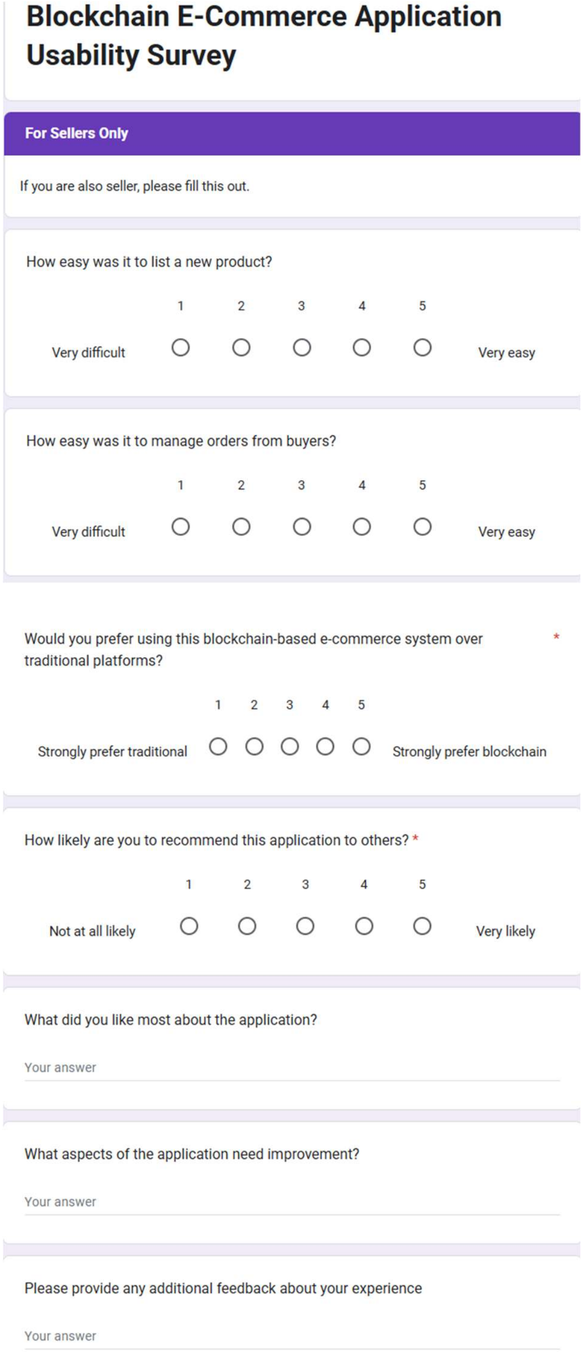
Task Completion

How easy was it to complete the checkout process? *

	1	2	3	4	5	
Very difficult	○	○	○	○	○	Very easy

How easy was it to track your order status? *

	1	2	3	4	5	
Very difficult	○	○	○	○	○	Very easy



Blockchain E-Commerce Application Usability Survey

For Sellers Only

If you are also seller, please fill this out.

How easy was it to list a new product?

	1	2	3	4	5	
Very difficult	○	○	○	○	○	Very easy

How easy was it to manage orders from buyers?

	1	2	3	4	5	
Very difficult	○	○	○	○	○	Very easy

Would you prefer using this blockchain-based e-commerce system over traditional platforms? *

	1	2	3	4	5	
Strongly prefer traditional	○	○	○	○	○	Strongly prefer blockchain

How likely are you to recommend this application to others? *

	1	2	3	4	5	
Not at all likely	○	○	○	○	○	Very likely

What did you like most about the application?

Your answer

What aspects of the application need improvement?

Your answer

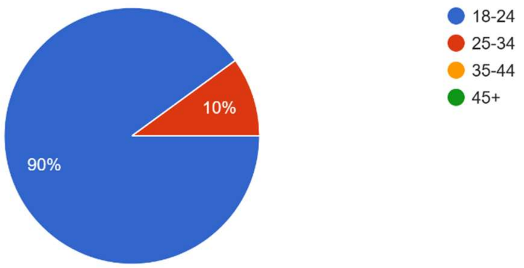
Please provide any additional feedback about your experience

Your answer

A.3 Survey Responses

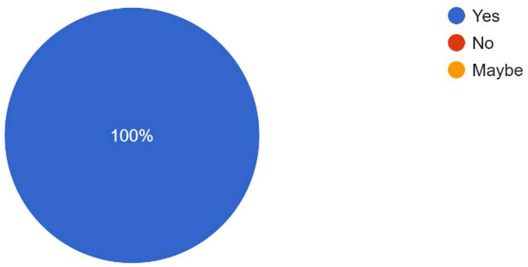
Age group

20 responses



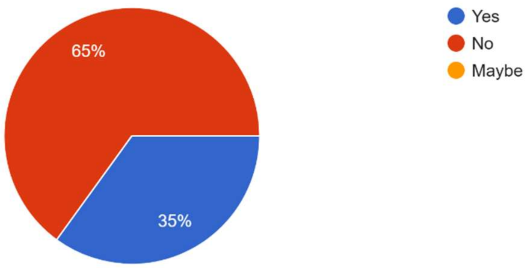
Have you used e-commerce platforms before? (e.g., Shopee, Lazada)

20 responses



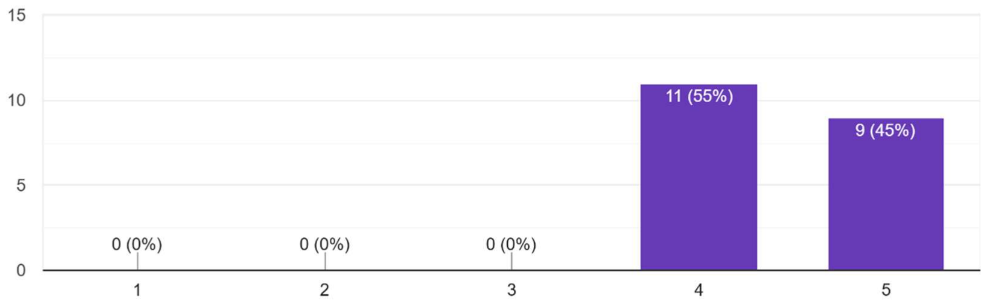
Have you used blockchain/cryptocurrency applications before?

20 responses



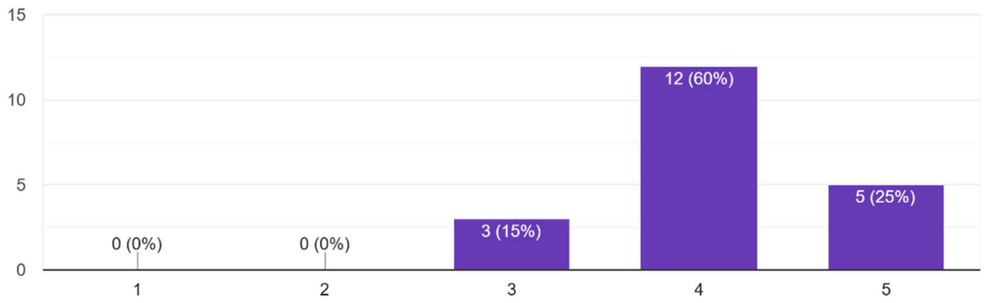
How easy was it to register and log into the application?

20 responses



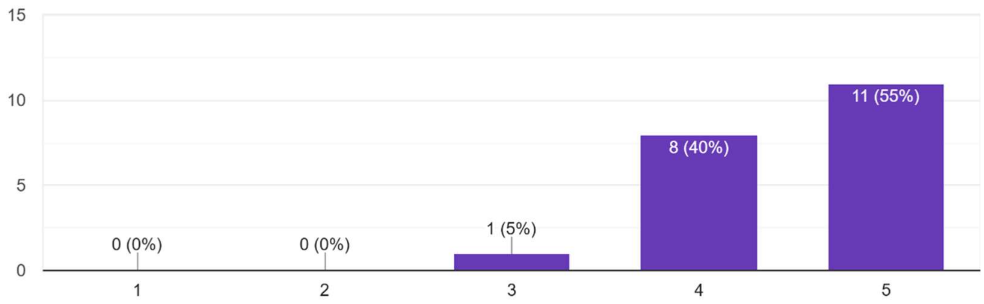
How easy was it to find products you were looking for?

20 responses



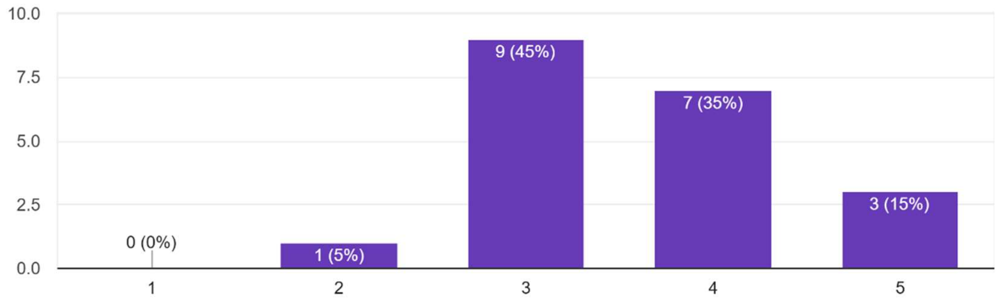
How would you rate the overall visual design of the application?

20 responses



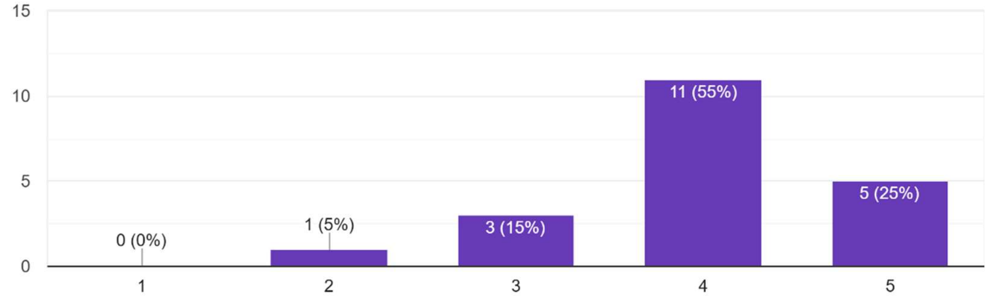
How easy was it to connect your cryptocurrency wallet?

20 responses



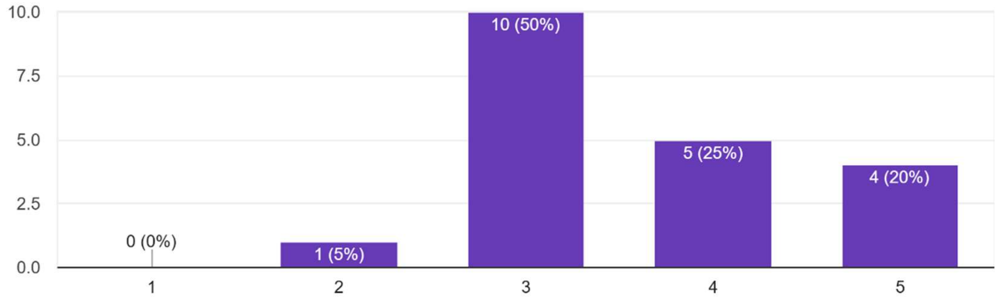
How clear was the information about blockchain transactions?

20 responses



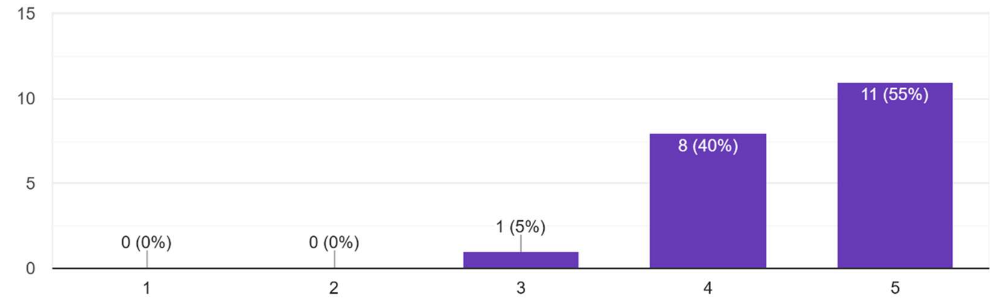
Did you understand the benefits of using blockchain for your purchase?

20 responses



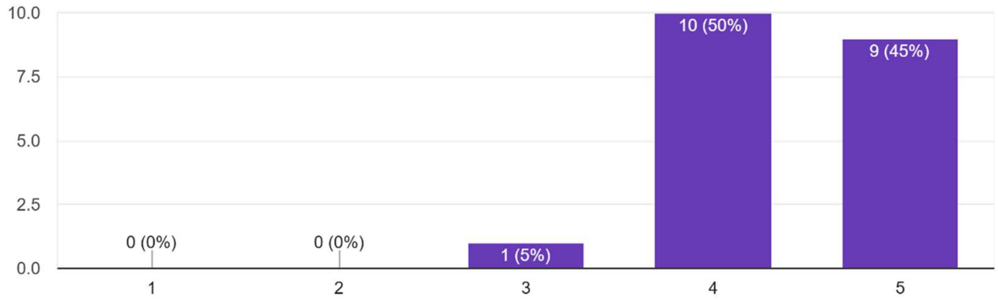
How easy was it to complete the checkout process?

20 responses



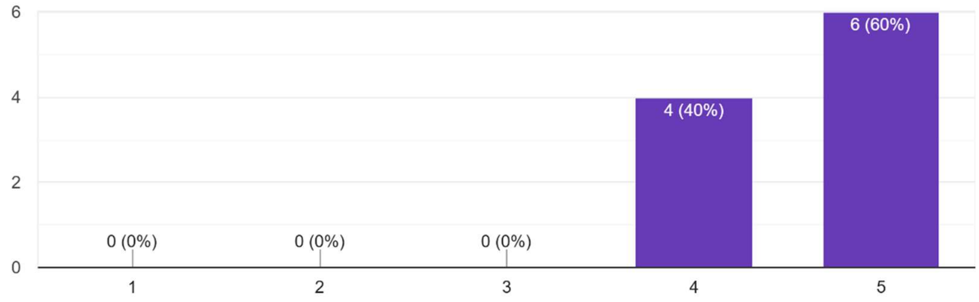
How easy was it to track your order status?

20 responses



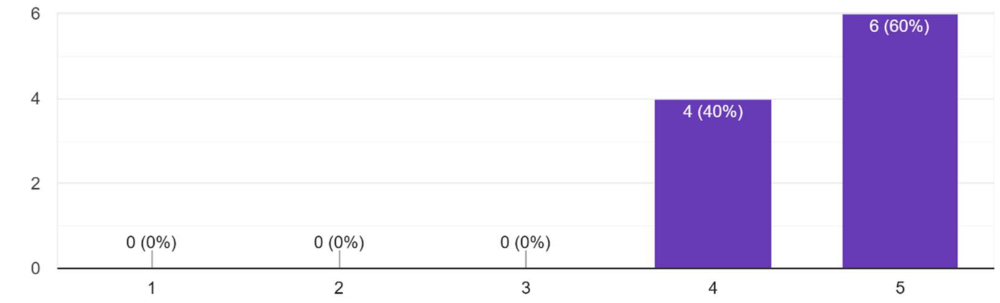
How easy was it to list a new product?

10 responses



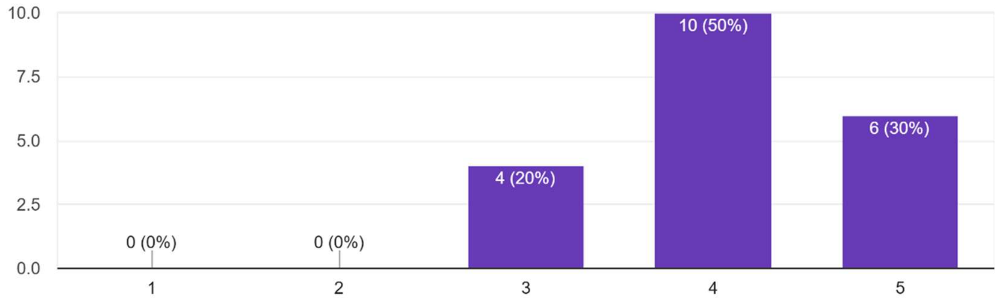
How easy was it to manage orders from buyers?

10 responses



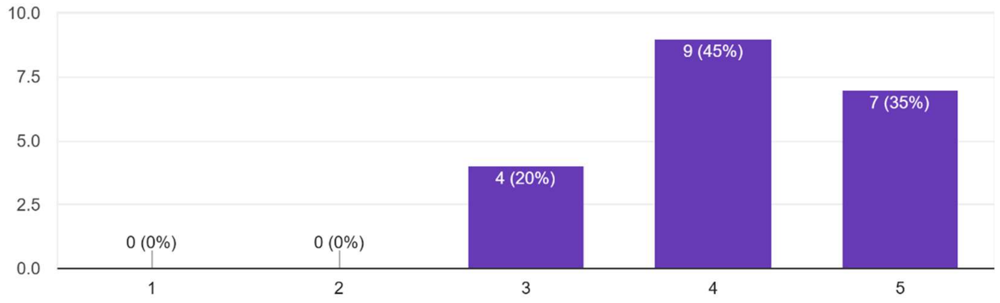
Would you prefer using this blockchain-based e-commerce system over traditional platforms?

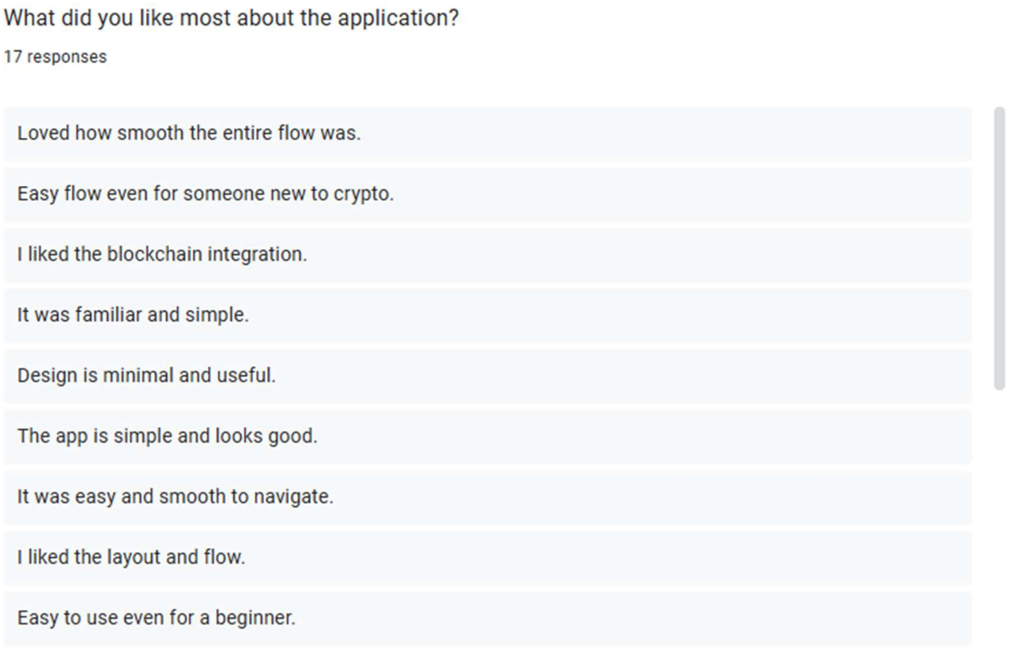
20 responses



How likely are you to recommend this application to others?

20 responses





What did you like most about the application?

17 responses

Loved how smooth the entire flow was.

Easy flow even for someone new to crypto.

I liked the blockchain integration.

It was familiar and simple.

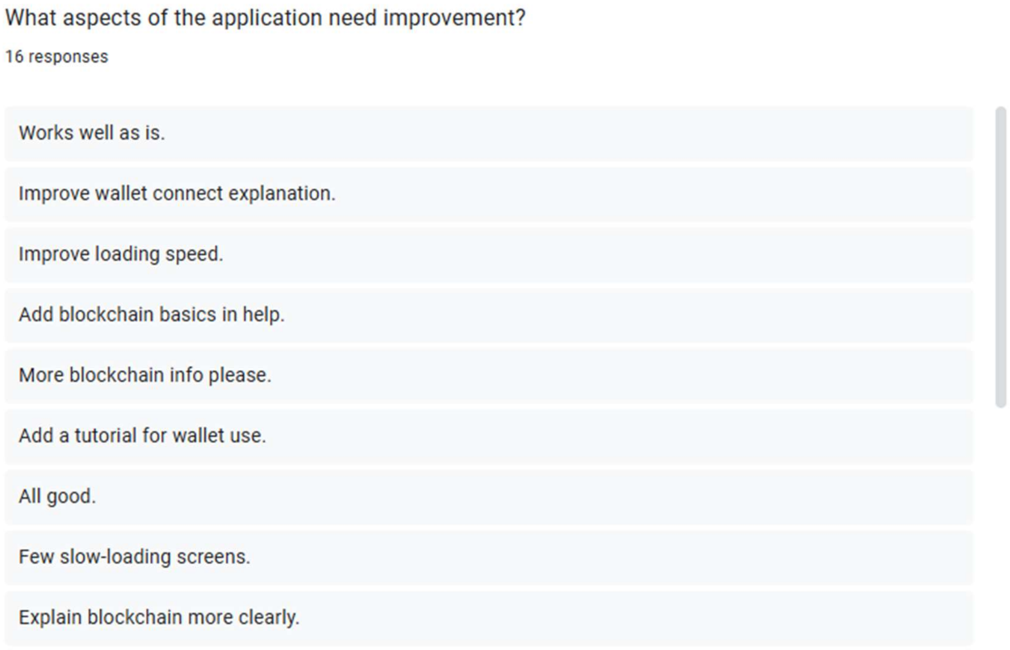
Design is minimal and useful.

The app is simple and looks good.

It was easy and smooth to navigate.

I liked the layout and flow.

Easy to use even for a beginner.



What aspects of the application need improvement?

16 responses

Works well as is.

Improve wallet connect explanation.

Improve loading speed.

Add blockchain basics in help.

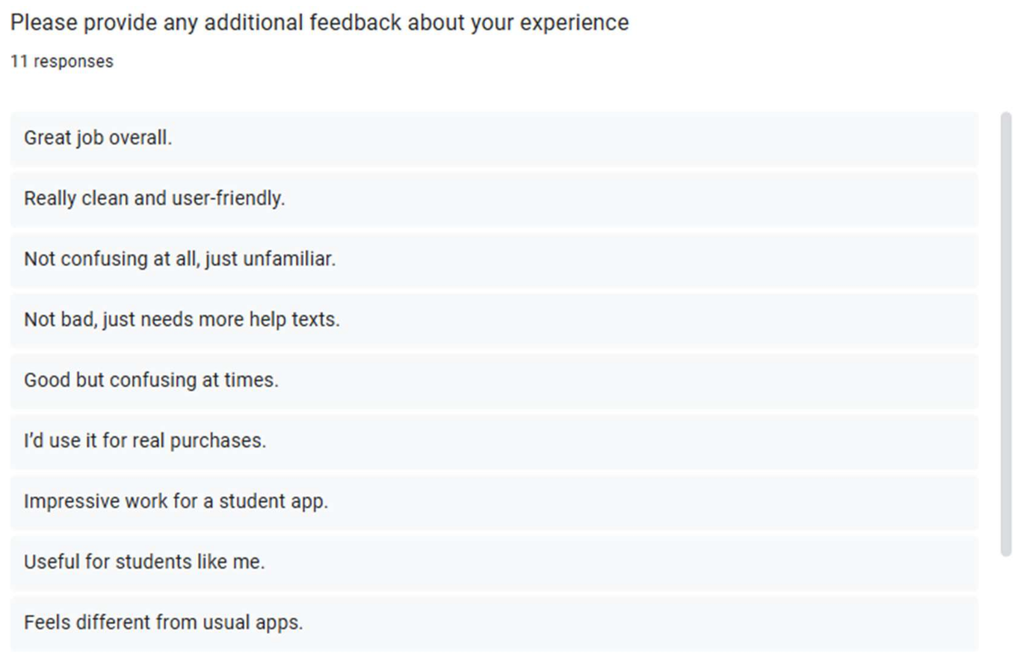
More blockchain info please.

Add a tutorial for wallet use.

All good.

Few slow-loading screens.

Explain blockchain more clearly.



Please provide any additional feedback about your experience

11 responses

Great job overall.

Really clean and user-friendly.

Not confusing at all, just unfamiliar.

Not bad, just needs more help texts.

Good but confusing at times.

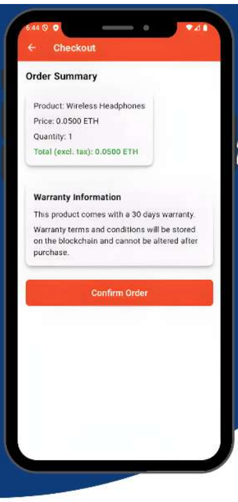
I'd use it for real purchases.

Impressive work for a student app.

Useful for students like me.

Feels different from usual apps.

A.4 Poster



Checkout

Order Summary

Product: Wireless Headphones
Price: 0.0500 ETH
Quantity: 1
Total (excl. tax): 0.0500 ETH

Warranty Information

This product comes with a 30 days warranty.

Warranty terms and conditions will be stored on the blockchain and cannot be altered after purchase.

Confirm Order

$0 FEE

DIRECT2U

PHYSICAL TO DIGITAL SERVICES PROFILING APP

A blockchain-based e-commerce application using Flutter to eliminate intermediaries in transactions, increasing trust and reducing costs.

PROBLEM IDENTIFIED

Current systems rely heavily on middlemen, **raising transaction costs** and **lowering transparency**.

TECHNOLOGIES USED

The project utilizes Flutter for the frontend, while the blockchain component is based on Ethereum, specifically using Ganache as local server. Smart contracts are developed in Solidity, and MetaMask is employed as the cryptocurrency wallet.

Discussion

The intuitive user interface and enhanced trust through transparency are significant strengths of the system, while the complexity of blockchain concepts poses a challenge for first-time users. Overall, the system effectively reduces costs and improves transactional transparency.

Conclusions

The project has successfully achieved its objectives. Looking ahead, the focus will be on incorporating blockchain tutorials and enhancing transaction speed.

Objectives

Deliver a user-friendly interface for registration, browsing, purchasing, and order management through the Ethereum blockchain.

Development

Designed the overall system architecture, coded and deployed smart contracts, integrated the blockchain with the user interface, and conducted demo user testing.

RESULT: USER FEEDBACK (Avg. Ratings out of 5)

- **Functionality:**
  - Registration (4.45/5), browsing (4.1/5), checkout (4.5/5), tracking (4.4/5).
- **Seller Tasks:**
  - Listing (4.6/5), management (4.6/5).
- **Satisfaction:**
  - Design (4.5/5)
- **Challenges:**
  - Wallet (3.6/5), blockchain clarity (3.6/5) for non-crypto users.

80%	Preferred over traditional platforms
80%	Would recommend to others

A UTAR UNIVERSITI TUNKU ABDUL RAHMAN FINAL YEAR PROJECT

PREPARED BY: TAN JUN JIE (22ACB02142)