VIETNAM-KOREA UNIVERSITY OF INFORMATION AND  
COMMUNICATION TECHNOLOGY

**Faculty of Computer Science**



GRADUATION THESIS

**RESEARCH AND APPLY BLOCKCHAIN TO**

**E-COMMERCE WEBSITE**

Student: **NGUYEN DAI NGO**

**VO NGUYEN NHAT TRUONG**

Class: **19IT5**

Supervisor: TS. HO VAN PHI

***Da Nang, December – 2023***

VIETNAM-KOREA UNIVERSITY OF INFORMATION AND

COMMUNICATION TECHNOLOGY

**Faculty of Computer Science**

GRADUATION THESIS

**RESEARCH AND APPLY BLOCKCHAIN TO E-COMMERCE WEBSITE**

Student: **NGUYEN DAI NGO**

**VO NGUYEN NHAT TRUONG**

Class: **19IT5**

Supervisor: TS. HO VAN PHI

***Da Nang, December - 2023***

SUPERVIOR’S COMMENT

ACKNOWLEDGEMENTS

We would like to express our heartfelt gratitude to those who have assisted us during our academic journey:

First and foremost, we would like to extend our sincerest thanks to Dr. Ho Van Phi, our thesis supervisor, who enthusiastically provided assistance, guidance, and clarification for queries throughout the thesis. We wish you and your family good health and success in your endeavors.

We deeply appreciate the professors at the University of Information Technology and Communication Viet – Han for imparting knowledge to us during the past semesters.

Finally, we would like to express our gratitude to all the teachers, friends who have helped and contributed to our completion of this graduation thesis, as our professional knowledge and objectivity are still limited. Therefore, the graduation thesis may have some shortcomings. We look forward to receiving your feedback and suggestions.

We sincerely thank you!

STATEMENT OF AUTHORSHIP

We hereby declare that the content of our project is completely original and we have not copied any part from any other source. We confirm that the ideas, text, images and any other materials presented in this project are the result of my own work, research and creativity.

I understand the importance of academic integrity and the consequences of plagiarism. Therefore, I commit that this project is the product of my own efforts and I take full responsibility for its authenticity. I have appropriately cited and referenced any outside sources I may have consulted during the research process.

*Danang, December 20th, 2023*

NGUYEN DAI NGO

**VO NGUYEN NHAT TRUONG**

TABLE OF CONTENTS

[SUPERVIOR’S COMMENT iv](#_Toc153488790)

[ACKNOWLEDGEMENTS v](#_Toc153488791)

[STATEMENT OF AUTHORSHIP i](#_Toc153488792)

[TABLE OF CONTENTS ii](#_Toc153488793)

[LIST OF FIGURES vii](#_Toc153488794)

[LIST OF TABLES x](#_Toc153488795)

[INTRODUCTION 1](#_Toc153488796)

[1. Problem statement 1](#_Toc153488797)

[2. Structure of the thesis 3](#_Toc153488798)

[CHAPTER 1. THEORETICAL BASIS 4](#_Toc153488799)

[1.1 Overview 4](#_Toc153488800)

[1.2 Learn about “ Research and apply Blockchain to e-commerce website ” 6](#_Toc153488801)

[1.2.1 Theory 6](#_Toc153488802)

[1.2.2 Function 6](#_Toc153488803)

[1.2.2.1 General function 6](#_Toc153488804)

[1.2.2.2 Administrator rights function 6](#_Toc153488805)

[1.2.2.3 User functions 7](#_Toc153488806)

[1.2.2.4 Sales function 7](#_Toc153488807)

[1.3 Tool 7](#_Toc153488808)

[1.4 Summary 7](#_Toc153488809)

[1.5 Build system 7](#_Toc153488810)

[1.5.1 Introducing Visual Studio code 7](#_Toc153488811)

[1.5.2 Introducing Xampp 8](#_Toc153488812)

[1.5.3 Metamask Wallet 8](#_Toc153488813)

[1.6 Other tools 9](#_Toc153488814)

[1.6.1 Introduction to ExpressJs Framework (Backend API) 9](#_Toc153488815)

[1.6.1.1 Overview of ExpressJs 9](#_Toc153488816)

[1.6.1.2 Benefits of using ExpressJs. 10](#_Toc153488817)

[1.6.2 Introduction to ReactJs Framework (Front End) 10](#_Toc153488818)

[1.6.2.1 Overview of ReactJs Framework 10](#_Toc153488819)

[1.6.2.2 Benefits of using ReactJs Framework 11](#_Toc153488820)

[1.6.3 Overview of Mysql database 11](#_Toc153488821)

[1.6.4 Overview of RESTful APIs 11](#_Toc153488822)

[1.6.5 Introducing the MVC (Model - View - Controller) model 12](#_Toc153488823)

[1.6.5.1 Overview of MVC 12](#_Toc153488824)

[1.6.5.2 Benefits of using the MVC model 12](#_Toc153488825)

[CHAPTER 2. SYSTEM ANALYSIS AND DESIGN 13](#_Toc153488826)

[2.1 Analyze project requirements 13](#_Toc153488827)

[2.1.1 Topic Description 13](#_Toc153488828)

[2.1.2 Topic Description 14](#_Toc153488829)

[2.2 Describes the entity's properties in detail 16](#_Toc153488830)

[2.2.1 AddressUser entity 16](#_Toc153488831)

[2.2.2 Allcode entity 16](#_Toc153488832)

[2.2.3 Banner entity 17](#_Toc153488833)

[2.2.4 Blog entity 17](#_Toc153488834)

[2.2.5 Message entity 18](#_Toc153488835)

[2.2.6 OrderDetail entity 18](#_Toc153488836)

[2.2.7 OrderProduct entity 19](#_Toc153488837)

[2.2.8 ProductDetail entity 19](#_Toc153488838)

[2.2.9 ProductDetailSize entity 20](#_Toc153488839)

[2.2.10 ProductImage entity 20](#_Toc153488840)

[2.2.11 Product entity 21](#_Toc153488841)

[2.2.12 RoomMessage entity 21](#_Toc153488842)

[2.2.13 ShopCart entity 22](#_Toc153488843)

[2.2.14 TypeShip entity 22](#_Toc153488844)

[2.2.15 TypeVoucher entity 23](#_Toc153488845)

[2.2.16 User entity 23](#_Toc153488846)

[2.2.17 Voucher entity 24](#_Toc153488847)

[2.2.18 VoucherUsed entity 24](#_Toc153488848)

[2.2.19 Comment entity 25](#_Toc153488849)

[2.2.20 Supplier entity 25](#_Toc153488850)

[2.2.21 Receipt entity 26](#_Toc153488851)

[2.2.22 ReceiptDetail entity 26](#_Toc153488852)

[2.3 E-R entity relationship symbol 27](#_Toc153488853)

[2.4 Database schema 27](#_Toc153488854)

[2.5 Data tables 28](#_Toc153488855)

[2.5.1 AddressUser table 28](#_Toc153488856)

[2.5.2 Allcode table 28](#_Toc153488857)

[2.5.3 Banner table 28](#_Toc153488858)

[2.5.4 Blog table 29](#_Toc153488859)

[2.5.5 Comment table 29](#_Toc153488860)

[2.5.6 Message table 30](#_Toc153488861)

[2.5.7 OrderDetail table 30](#_Toc153488862)

[2.5.8 OrderProduct table 30](#_Toc153488863)

[2.5.9 ProductDetail table 31](#_Toc153488864)

[2.5.10 ProductDetailSize table 31](#_Toc153488865)

[2.5.11 ProductImage table 31](#_Toc153488866)

[2.5.12 Product table 32](#_Toc153488867)

[2.5.13 RoomMessage table 32](#_Toc153488868)

[2.5.14 ShopCart table 33](#_Toc153488869)

[2.5.15 TypeShip table 33](#_Toc153488870)

[2.5.16 TypeVoucher table 33](#_Toc153488871)

[2.5.17 User table 34](#_Toc153488872)

[2.5.18 Voucher table 35](#_Toc153488873)

[2.5.19 VoucherUser table 35](#_Toc153488874)

[2.5.20 Supplier table 35](#_Toc153488875)

[2.5.21 Receipt table 36](#_Toc153488876)

[2.5.22 ReceiptDetail table 36](#_Toc153488877)

[2.6 Actors in the request 37](#_Toc153488878)

[2.7 Use Cases in the system 37](#_Toc153488879)

[2.7.1 General Use cases 38](#_Toc153488880)

[2.7.2 Use Case of admin 38](#_Toc153488881)

[2.7.3 Use Case of user 41](#_Toc153488882)

[2.7.4 Use Case of a visitor 43](#_Toc153488883)

[2.7.5 Use case of a salesperson 43](#_Toc153488884)

[2.8 List of actors 44](#_Toc153488885)

[2.9 List of Use cases 45](#_Toc153488886)

[2.10 Use Case diagram based on each actor detail 49](#_Toc153488887)

[2.11 Specification of Use Cases 51](#_Toc153488888)

[2.11.1 Use case for visiting guests 51](#_Toc153488889)

[2.11.2 Use case for visiting guests 52](#_Toc153488890)

[2.11.3 Use case Admin 54](#_Toc153488891)

[2.11.4 Use case Saler 57](#_Toc153488892)

[2.12 Activity diagram 58](#_Toc153488893)

[2.12.1 Admin activity diagram 58](#_Toc153488894)

[2.12.2 User activity diagram 61](#_Toc153488895)

[2.12.3 Viewer activity diagram 63](#_Toc153488896)

[2.12.4 Activity diagram of sales staff 64](#_Toc153488897)

[2.13 Algorithm diagram 65](#_Toc153488898)

[2.13.1 Login 65](#_Toc153488899)

[2.13.2 Log out 65](#_Toc153488900)

[2.13.3 Delete data 66](#_Toc153488901)

[2.13.4 Updating data 66](#_Toc153488902)

[2.13.5 Add data 67](#_Toc153488903)

[2.13.6 Cart 67](#_Toc153488904)

[CHAPTER 3. SETUP AND TEST RESULTS 68](#_Toc153488905)

[3.1 Software requirements 68](#_Toc153488906)

[3.2 Design the theme 68](#_Toc153488907)

[3.2.1 List of interfaces 68](#_Toc153488908)

[3.3 Design the theme 69](#_Toc153488909)

[CONCLUSIONS AND SUGGESTIONS 71](#_Toc153488910)

[1. Conclusions 71](#_Toc153488911)

[2. Suggestions 71](#_Toc153488912)

[REFERENCES 72](#_Toc153488913)

[PUBLICATIONS 73](#_Toc153488914)

[APPENDIX 74](#_Toc153488915)

LIST OF FIGURES

[Figure 1. 1 - Visual Studio Code 15](#_Toc153819115)

[Figure 1. 2 – Xampp 16](#_Toc153819116)

[Figure 1. 3 - MetaMask Wallet 16](#_Toc153819117)

[Figure 1. 4 - Structure of ExpressJs 18](#_Toc153819118)

[Figure 2. 1 - Address User entity 26](#_Toc153819122)

[Figure 2. 2 - Allcode entity 26](#_Toc153819123)

[Figure 2. 3 - Banner entity 27](#_Toc153819124)

[Figure 2. 4 - Blog entity 27](#_Toc153819125)

[Figure 2. 5 - Message entity 28](#_Toc153819126)

[Figure 2. 6 - OrderDetail entity 28](#_Toc153819127)

[Figure 2. 7 - OrderProduct entity 29](#_Toc153819128)

[Figure 2. 8 - ProductDetail entity 29](#_Toc153819129)

[Figure 2. 9 - ProductDetailSize entity 30](#_Toc153819130)

[Figure 2. 10 - ProductImage entity 30](#_Toc153819131)

[Figure 2. 11 - Product entity 31](#_Toc153819132)

[Figure 2. 12 - RoomMessage entity 31](#_Toc153819133)

[Figure 2. 13 - ShopCart entity 32](#_Toc153819134)

[Figure 2. 14 - TypeShip entity 32](#_Toc153819135)

[Figure 2. 15 - TypeVoucher entity 33](#_Toc153819136)

[Figure 2. 16 - User entity 33](#_Toc153819137)

[Figure 2. 17 - Voucher entity 34](#_Toc153819138)

[Figure 2. 18 - VoucherUsed entity 34](#_Toc153819139)

[Figure 2. 19 - Comment entity 35](#_Toc153819140)

[Figure 2. 20 - Supplier entity 35](#_Toc153819141)

[Figure 2. 21 - Receipt entity 36](#_Toc153819142)

[Figure 2. 22 - ReceiptDetail entity 36](#_Toc153819143)

[Figure 2. 23 - E-R entity relationship symbol 37](#_Toc153819144)

[Figure 2. 24 - Database diagram 37](#_Toc153819145)

[Figure 2. 25 - Use Case Admin 47](#_Toc153819146)

[Figure 2. 26 - Use Case User 47](#_Toc153819147)

[Figure 2. 27 - Usercase Saler 47](#_Toc153819148)

[Figure 2. 28 - Use Case Viewer 47](#_Toc153819149)

[Figure 2. 29 - General use cases 48](#_Toc153819150)

[Figure 2. 30 - Use Case User management 48](#_Toc153819151)

[Figure 2. 31 - Use Case Product Management 49](#_Toc153819152)

[Figure 2. 32 - Use Case Brand management 49](#_Toc153819153)

[Figure 2. 33 - Use Case Manage posts 49](#_Toc153819154)

[Figure 2. 34 - Use Case Promotion management 49](#_Toc153819155)

[Figure 2. 35 - Use Case Manage promotion types 49](#_Toc153819156)

[Figure 2. 36 - Use Case Manage shipping types 50](#_Toc153819157)

[Figure 2. 37 - Use Case Order management 50](#_Toc153819158)

[Figure 2. 38 - Use case Topic management 50](#_Toc153819159)

[Figure 2. 39 - Use case Banner management 50](#_Toc153819160)

[Figure 2. 40 - Use case Manage messages 50](#_Toc153819161)

[Figure 2. 41 - Use case Supplier management 50](#_Toc153819162)

[Figure 2. 42 - Use case Management of import goods 51](#_Toc153819163)

[Figure 2. 43 - Use case Statistical 51](#_Toc153819164)

[Figure 2. 44 - Use case Manage personal information 51](#_Toc153819165)

[Figure 2. 45 - Use case Manage personal orders 51](#_Toc153819166)

[Figure 2. 46 - Use case Manage personal addresses 51](#_Toc153819167)

[Figure 2. 47 - Use case Manage personal promotions 52](#_Toc153819168)

[Figure 2. 48 - Use case Add product 52](#_Toc153819169)

[Figure 2. 49 - Use case Register an account 52](#_Toc153819170)

[Figure 2. 50 - Use case Oder payment 52](#_Toc153819171)

[Figure 2. 51 - Use case Comments, reviews 52](#_Toc153819172)

[Figure 2. 52 - Use case View products 52](#_Toc153819173)

[Figure 2. 53 - Use case View news posts 53](#_Toc153819174)

[Figure 2. 54 - Use case View products 53](#_Toc153819175)

[Figure 2. 55 - Use case View news posts 53](#_Toc153819176)

[Figure 2. 56 - Use case Register 53](#_Toc153819177)

[Figure 2. 57 - Use case Login 53](#_Toc153819178)

[Figure 2. 58 - Use case Order management 53](#_Toc153819179)

[Figure 2. 59 - Use case Supplier management 54](#_Toc153819180)

[Figure 2. 60 - Use case Management of import goods 54](#_Toc153819181)

[Figure 2. 61 - Use case Messages 54](#_Toc153819182)

[Figure 2. 62 - Use case diagram according to the actor is Admin 59](#_Toc153819183)

[Figure 2. 63 - Use case diagram for each User actor 60](#_Toc153819184)

[Figure 2. 64 - Use case diagram for each guest actor 60](#_Toc153819185)

[Figure 2. 65 - Usecase diagram for each salesperson 61](#_Toc153819186)

[Figure 2. 66 - Diagram of admin login activities 69](#_Toc153819187)

[Figure 2. 67 - User management activity diagram 69](#_Toc153819188)

[Figure 2. 68 - Activity diagram for managing all categories 70](#_Toc153819189)

[Figure 2. 69 - Statistical activity diagram 70](#_Toc153819190)

[Figure 2. 70 - Diagram of personal information management activities 71](#_Toc153819191)

[Figure 2. 71 - Diagram of personal order management activities 71](#_Toc153819192)

[Figure 2. 72 - Activity diagram of adding products to cart 72](#_Toc153819193)

[Figure 2. 73 - Personal address management diagram 72](#_Toc153819194)

[Figure 2. 74 - Diagram of account registration activities 73](#_Toc153819195)

[Figure 2. 75 - Activity diagram for viewing product information 73](#_Toc153819196)

[Figure 2. 76 - Activity diagram for viewing post information 73](#_Toc153819197)

[Figure 2. 77 - Message management activity diagram 74](#_Toc153819198)

[Figure 2. 78 - Diagram of import management activities 74](#_Toc153819199)

[Figure 2. 79 - Login flow diagram 75](#_Toc153819200)

[Figure 2. 80 - Logout flow diagram 75](#_Toc153819201)

[Figure 2. 81 - Data deletion flow diagram 76](#_Toc153819202)

[Figure 2. 82 - Data update flow diagram 76](#_Toc153819203)

[Figure 2. 83 - Flow diagram for adding data 77](#_Toc153819204)

[Figure 2. 84 - Flow diagram of adding products to cart 77](#_Toc153819205)

[Figure 3. 1 - Home page 80](#_Toc153819206)

[Figure 3. 2 - Product portfolio page 80](#_Toc153819207)

[Figure 3. 3 - News page 81](#_Toc153819208)

[Figure 3. 4 - Voucher page 81](#_Toc153819209)

[Figure 3. 5 - Product details page 82](#_Toc153819210)

[Figure 3. 6 - Cart page 82](#_Toc153819211)

[Figure 3. 7 - Checkout page 83](#_Toc153819212)

[Figure 3. 8 - Payment page using VnPay 83](#_Toc153819213)

[Figure 3. 9 - Placed order page 84](#_Toc153819214)

[Figure 3. 10 - Personal information page 84](#_Toc153819215)

[Figure 3. 11 - User management page 85](#_Toc153819216)

[Figure 3. 12 - Category management page 85](#_Toc153819217)

[Figure 3. 13 - Brand management page 86](#_Toc153819218)

[Figure 3. 14 - Product management page 86](#_Toc153819219)

[Figure 3. 15 - Banner management page 87](#_Toc153819220)

[Figure 3. 16 - Theme management page 87](#_Toc153819221)

[Figure 3. 17 - Shiping type management page 88](#_Toc153819222)

[Figure 3. 18 - Supplier management page 88](#_Toc153819223)

[Figure 3. 19 - Inventory management page 89](#_Toc153819224)

[Figure 3. 20 - Order management page 89](#_Toc153819225)

LIST OF TABLES

[Table 2. 1 - AddressUser table 28](#_Toc153488606)

[Table 2. 2 - Allcode table 28](#_Toc153488607)

[Table 2. 3 - Banner table 29](#_Toc153488608)

[Table 2. 4 - Blog table 29](#_Toc153488609)

[Table 2. 5 - Comment table 29](#_Toc153488610)

[Table 2. 6 - Message table 30](#_Toc153488611)

[Table 2. 7 - OrderDetail table 30](#_Toc153488612)

[Table 2. 8 - OrderProduct table 31](#_Toc153488613)

[Table 2. 9 - ProductDetail table 31](#_Toc153488614)

[Table 2. 10 - ProductDetailSize table 31](#_Toc153488615)

[Table 2. 11 - ProductImage table 32](#_Toc153488616)

[Table 2. 12 - Product table 32](#_Toc153488617)

[Table 2. 13 - RoomMessage table 33](#_Toc153488618)

[Table 2. 14 - ShopCart table 33](#_Toc153488619)

[Table 2. 15 - TypeShip table 33](#_Toc153488620)

[Table 2. 16 - TypeVoucher table 34](#_Toc153488621)

[Table 2. 17 - User table 34](#_Toc153488622)

[Table 2. 18 - Voucher table 35](#_Toc153488623)

[Table 2. 19 - VoucherUser table 35](#_Toc153488624)

[Table 2. 20 - Supplier table 36](#_Toc153488625)

[Table 2. 21 - Receipt table 36](#_Toc153488626)

[Table 2. 22 - ReceiptDetail table 36](#_Toc153488627)

[Table 2. 23 - List of actors 45](#_Toc153488628)

[Table 2. 24 - List of Use cases 48](#_Toc153488629)

[Table 3. 1 - Interface list table 69](#_Toc153488647)

INTRODUCTION

##### Problem statement

In the ever-evolving landscape of the fashion industry, staying ahead requires more than just an eye for the latest trends; it demands a cutting-edge approach to business operations. Welcome to the future of fashion retail, where the fusion of technology and style takes center stage. In this article, we delve into the transformative journey of building a Fashion Store Management Website integrated with Blockchain technology—a game-changer that promises to redefine the way we perceive and manage fashion businesses.

The Promise of Blockchain Integration:

At the heart of this revolution is blockchain technology—a decentralized and secure ledger system that has proven its mettle in various industries. By integrating blockchain into the core of a fashion store management website, we unlock a myriad of possibilities that address key challenges faced by the industry.

Transparency and Traceability:

Blockchain ensures an immutable record of transactions, providing an unprecedented level of transparency in the fashion supply chain. From sourcing materials to production and distribution, every step is traceable, fostering accountability and ethical practices.

Security Reinvented:

The fashion industry, like any other, grapples with cybersecurity threats and data breaches. Blockchain's cryptographic principles create a fortress of security around sensitive data, protecting customer information, transaction records, and intellectual property from potential threats.

Smart Contracts for Seamless Operations:

Smart contracts, self-executing agreements with the terms of the contract directly written into code, automate various processes in the fashion store management ecosystem. From inventory management to order fulfillment, these contracts streamline operations, reduce human error, and enhance efficiency.

**Enhanced Customer Trust:**

Consumers are becoming more conscious about the origins of their fashion choices. Blockchain integration allows customers to verify the authenticity and sustainability of products, fostering trust and loyalty. This transparency can be a powerful differentiator in a competitive market.

**Key Components of Development:**

User-Friendly Interface:

Crafting an intuitive and visually appealing interface is paramount. A user-friendly design ensures that all stakeholders, from administrators to end customers, can navigate the platform effortlessly.

Blockchain Infrastructure:

Selecting the right blockchain infrastructure is critical. Whether utilizing public blockchains like Ethereum or implementing a private blockchain solution, understanding the unique requirements of the fashion industry is key to successful integration.

Smart Contract Implementation:

Developing and deploying smart contracts tailored to the specific needs of the fashion store is crucial. These contracts should automate processes such as order processing, inventory updates, and promotions, enhancing operational efficiency.

**Conclusion:**

As we embark on the journey of building a Fashion Store Management Website with Blockchain integration, we step into a future where fashion and technology coalesce seamlessly. This innovative approach promises to revolutionize the industry, offering unprecedented transparency, security, and efficiency. By embracing the power of blockchain, fashion retailers can not only keep pace with the evolving market but also set new standards for sustainable and ethical business practices. Aims and Objectives

**Purpose:**

The primary purpose of this project is to develop a cutting-edge E-commerce Website with Blockchain integration. By harnessing the power of blockchain technology, we aim to revolutionize the traditional e-commerce landscape, introducing a more secure, transparent, and efficient platform for online transactions.

**Goals:**

**Enhanced Security:**

Implement blockchain's cryptographic features to fortify the security of user data, financial transactions, and sensitive information.

Establish a tamper-resistant system that ensures the integrity of user data and transaction records.

**Decentralized Transactions:**

Facilitate peer-to-peer transactions without the need for intermediaries, reducing transaction costs and increasing transaction speed.

Leverage blockchain's decentralized nature to create a trustless and resilient transaction system.

**Smart Contracts Implementation:**

Utilize smart contracts to automate and self-execute agreements between buyers and sellers, ensuring fair and transparent transactions.

Implement smart contracts to govern various aspects of the e-commerce process, such as order fulfillment, payment processing, and dispute resolution.

**Supply Chain Transparency:**

Utilize blockchain to create an immutable and transparent ledger for the entire supply chain, allowing users to trace the origin and journey of products.

Enhance visibility into the supply chain, providing consumers with detailed information about product authenticity and sourcing.

**Cryptocurrency Payments:**

Enable cryptocurrency payments to offer users an alternative payment method, expanding the customer base to a global audience.

Explore the integration of various cryptocurrencies to provide flexibility in payment options.

**Immutable Product Records:**

Store product information on the blockchain to ensure the immutability of details such as product specifications, origin, and history.

Increase trust among consumers by providing a secure and unalterable record of product data.

**User Experience and Interface:**

Design an intuitive and user-friendly interface for a seamless online shopping experience.

Prioritize user experience to ensure that blockchain integration does not compromise the ease of navigation and accessibility.

**Feedback and Iteration:**

Gather user feedback to continuously improve and refine the platform.

Iterate on the website based on user suggestions and emerging technologies in the blockchain and e-commerce domains.

By achieving these goals, the project aims to contribute to the evolution of e-commerce by building a secure, transparent, and efficient online marketplace powered by blockchain technology.

##### Structure of the thesis

After the *Introduction*, the thesis is structured in three chapters:

*Chapter 1,* **Theoretical basis**

*Chapter 2,* **System analysis and design**

*Chapter 3,* **Setup and test results**

Finally, there are *Conclusions*, *Suggestions,* *References* and *Appendices* related to the topic.

# THEORETICAL BASIS

## Overview

The integration of blockchain technology into e-commerce websites represents a groundbreaking approach to address inherent challenges in traditional online marketplaces. By combining the decentralized and secure features of blockchain with the dynamics of e-commerce, businesses can enhance transparency, security, and overall efficiency in online transactions.

**Decentralization:**

Blockchain introduces a decentralized structure, eliminating the need for a central authority or intermediary in e-commerce transactions.

Peer-to-peer transactions become feasible, reducing dependence on traditional payment processors and minimizing associated costs.

**Smart Contracts:**

Smart contracts, self-executing pieces of code, automate and enforce contractual agreements between buyers and sellers.

They streamline processes such as payment, order fulfillment, and dispute resolution, enhancing the efficiency of e-commerce operations.

**Security Enhancements:**

Cryptographic features of blockchain ensure a secure and tamper-resistant environment for user data, transactions, and sensitive information.

Enhanced security measures protect against fraud, data breaches, and unauthorized access, fostering trust among users.

**Supply Chain Transparency:**

Blockchain's transparent and immutable ledger facilitates end-to-end visibility in the supply chain.

Consumers can trace the origin, authenticity, and journey of products, promoting trust and ethical sourcing.

**Cryptocurrency Integration:**

E-commerce platforms can accept cryptocurrency payments, expanding payment options for users.

Cryptocurrencies enable cross-border transactions, reducing international payment complexities and transaction fees.

**Immutable Product Records:**

Product information stored on the blockchain ensures data integrity, preventing tampering or misinformation.

Immutable records contribute to consumer confidence by providing verifiable details about products.

**User Experience Considerations:**

Designing user-friendly interfaces that seamlessly integrate blockchain features without compromising the ease of use for customers.

Prioritizing a positive and intuitive user experience to encourage broader adoption of blockchain-based e-commerce platforms.

**Benefits:**

**Trust and Transparency:**

Blockchain instills trust through transparent and verifiable transactions, reducing the risk of fraud and enhancing consumer confidence.

**Reduced Cost:**

Elimination of intermediaries and streamlined processes lead to cost reductions in transaction fees and operational expenses.

**Global Reach:**

Cryptocurrency integration allows for global transactions, expanding the potential customer base beyond geographical boundaries.

**Data Security:**

Blockchain's cryptographic features ensure the security and integrity of user data, protecting against unauthorized access and data breaches.

Challenges and Considerations:

**Scalability:**

Addressing scalability issues to accommodate the growing number of transactions on blockchain networks.

**Regulatory Landscape:**

Navigating the evolving regulatory environment surrounding cryptocurrencies and blockchain technology.

**Education and Adoption:**

Educating users and businesses about blockchain technology to promote widespread adoption.

**Conclusion:**

Building e-commerce websites with blockchains offers a transformative approach to online transactions, introducing unprecedented levels of security, transparency, and efficiency. As businesses explore and implement blockchain solutions, the landscape of e-commerce is poised for significant advancements, promising a more secure and user-centric online shopping experience.

The issue of building an online store is highly essential. This helps the store in better managing its inventory, quickly and accurately generating revenue reports, reducing manpower, cutting costs, and saving time. Advertising products and brand promotion also become more accessible. From the customer's perspective, an online store helps them save shopping time and allows them to "reserve" their favorite products in advance, without the fear of items being purchased or sold out when they are not physically present at the store. It empowers them to actively browse and place orders.

Considering the mentioned aspects, we have chosen the topic of " **Research and apply Blockchain to e-commerce** **website** " to assist a fashion store in improving its inventory management and product advertising. This choice aligns with the current global trend of development.

## Learn about “ Research and apply Blockchain to e-commerce website ”

### Theory

Learn about:

* Blockchains
* Smart contracts
* MetaMask Wallet
* Learn operations related to sales websites (Shopping cart, ordering, review...)
* Learn the RESTful API model.
* Front end: Html, CSS, ReactJs, Redux
* Backend: Nodejs, ExpressJs
* Learn about online payment with PayPal Sandbox, VNPAY
* Learn OTP authentication with firebase
* Learn API authentication with JWT (Json web token)
* Learn online messaging (Socket io)
* Learn to log in to google Facebook (auth firebase)
* Learn to send email and edit text (markdown)

### Function

#### General function

* Register, log in, log out, forgot password, view and change personal information, search, text messages.

#### Administrator rights function

* Manage users, categories, products, orders, brands, banners, topics, posts, shipping types, promotions, orders, suppliers, import goods, messages, statistics.

#### User functions

* View product details, Add/Remove products in cart
* Order, Pay online, View order details, news posts, purchase history, Comments, product reviews.

#### Sales function

* Manage orders, suppliers, import goods, messages

## Tool

* Xampp
* Visual Studio Code
* MetaMask Wallet

## Summary

* Clearly understand and apply Blockchain to e-commerce
* Helps employees manage store activities, manage categories, view reports and statistics.
* Ensure customers have accurate product information and simple ordering and payment.

## Build system

### Introducing Visual Studio code

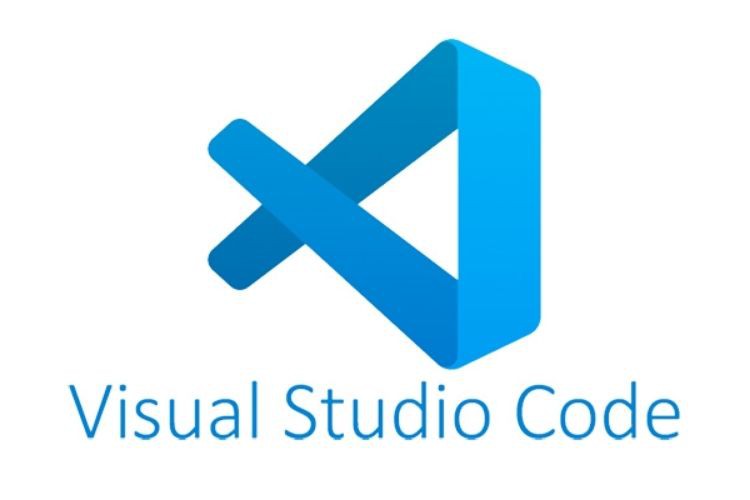


Figure 1. - Visual Studio Code

As a free programming code editor for Windows, Linux, and macOS, Visual Studio Code is developed by Microsoft. It is considered a perfect blend of an Integrated Development Environment (IDE) and a Code Editor. Visual Studio Code supports programming in various languages such as C/C++, F#, C#, and more. It includes debugging capabilities, comes with Git integration, features syntax highlighting, intelligent code completion, snippets, and source code enhancements.

Thanks to its customizable features, Visual Studio Code allows users to modify themes, shortcuts, and various options. It supports a wide range of web applications and also features a text editor and website design capabilities.

### Introducing Xampp



Figure 1. – Xampp

Xampp software is one of the widely used tools by many programmers for setting up websites using the PHP language. XAMPP serves the purpose of establishing a web server with pre-installed tools such as PHP, Apache, MySQL, etc. Xampp features a user-friendly interface, allowing programmers to easily enable, disable, or reboot server features at any time. Additionally, Xampp is built on open-source code.

Overall, Xampp is considered a crucial software tool for PHP programmers in the website development domain. Xampp installs essential components that complement each other, including:

* Apache
* PHP (providing the platform for \*.php script files to function)
* MySQL (database management system)
* Perl

### Metamask Wallet

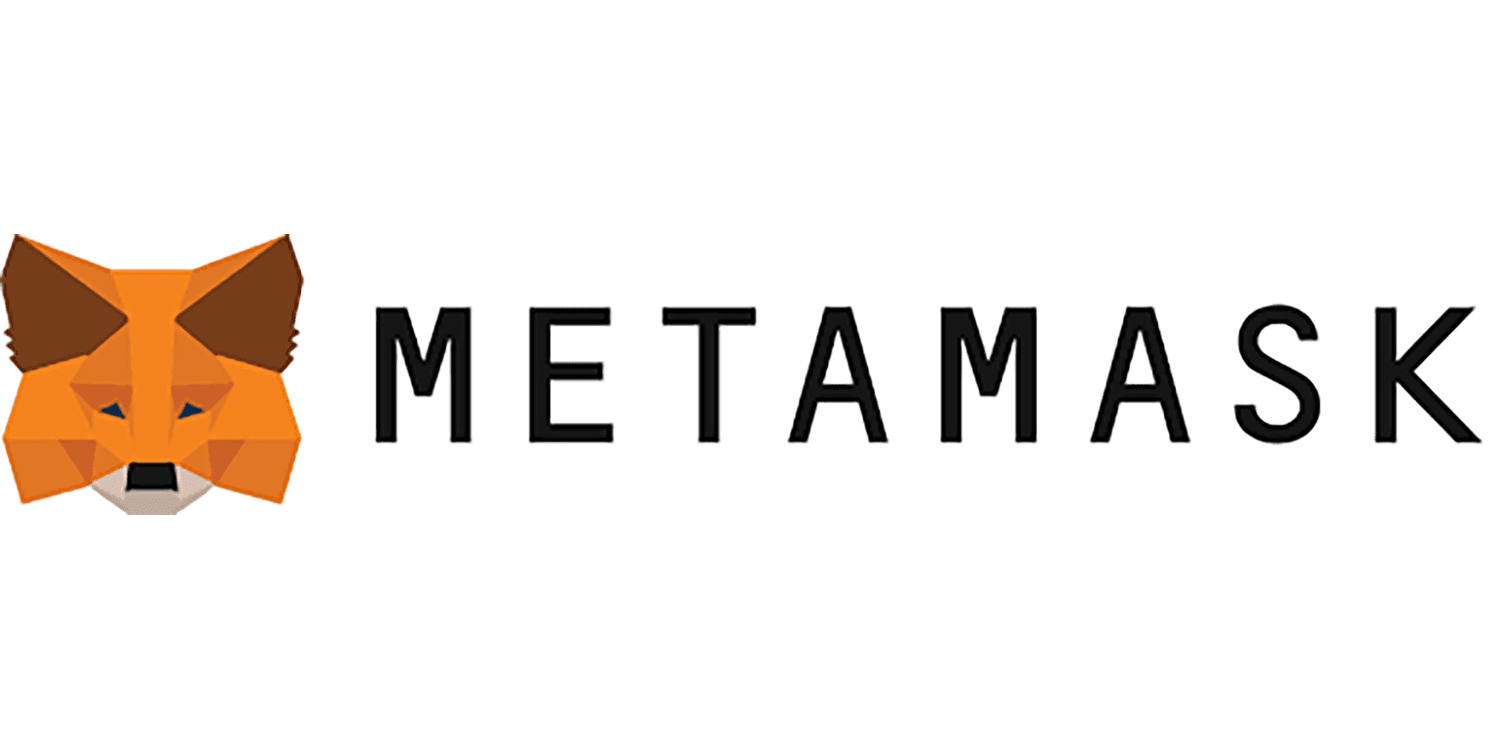


Figure 1. - MetaMask Wallet

MetaMask is a cryptocurrency wallet and browser extension that allows users to interact with decentralized applications (DApps) on the Ethereum blockchain. It serves as a bridge between your web browser and the Ethereum blockchain, enabling you to manage your Ethereum-based assets, including cryptocurrencies and non-fungible tokens (NFTs).

Here are some key features and aspects of the MetaMask wallet:

Wallet and Key Management: MetaMask provides users with a secure wallet to store their Ethereum and ERC-20 tokens. It also manages the private keys needed to access and authorize transactions on the Ethereum network.

Browser Extension: MetaMask operates as a browser extension for popular browsers like Google Chrome, Mozilla Firefox, and others. This extension allows users to seamlessly interact with decentralized applications directly from their web browser.

User-Friendly Interface: MetaMask has a user-friendly interface that simplifies the process of managing cryptocurrencies and interacting with DApps. It provides a straightforward way to send and receive Ethereum and tokens.

Decentralized Application (DApp) Support: Metamask facilitates the interaction with Ethereum-based DApps. When you visit a website that is a DApp, Metamask can detect it and provide you with a user interface to connect your wallet and engage with the application.

Secure and Private: MetaMask takes measures to ensure the security and privacy of users' private keys. It uses encryption techniques to protect sensitive information and allows users to set up a secure password.

Network Customization: Users can connect MetaMask to different Ethereum networks, including the Ethereum main net, various testnets, and custom networks. This feature is useful for developers and users testing applications on different environments.

Transaction Confirmation: Before any transaction is executed, Metamask prompts users to review and confirm the details, providing an additional layer of security to prevent accidental or unauthorized transactions.

MetaMask has become a popular choice among cryptocurrency users and developers for its ease of use, compatibility with various DApps, and its role in facilitating a seamless experience in the decentralized finance (DeFi) space.

## Other tools

### Introduction to ExpressJs Framework (Backend API)

#### Overview of ExpressJs

Express.js is a framework built on the Node.js platform. It provides powerful features for developing web or mobile applications. Express.js supports HTTP methods and middleware, creating a robust and user-friendly API.

Structure of Express Js :

Description: Diagram

Description automatically generated

Figure 1. - Structure of ExpressJs

Root:

* app.js contains information about configuration, declarations, definitions,... so that our application runs ok.
* package.json contains the packages for the application to run. If you have worked with PHP or RoR, this file has the same function as composer.json or Gemfile
* Folder routes contains routes in the application
* Folder view contains view/template for the application
* The public folder contains css, js, images,... files for the application

#### Benefits of using ExpressJs.

* Very easy to learn, requiring only knowledge of JavaScript; you won't need to learn a new language to pick up ExpressJS.
* Facilitates much easier backend development when utilizing ExpressJS.
* JavaScript code is interpreted through the Google V8 JavaScript Engine of Node.js. As a result, the code is executed quickly and effortlessly.
* ExpressJS is highly straightforward to customize and use according to specific needs.
* Provides a flexible and highly useful middleware module for performing additional tasks based on feedback and requests.
* Supports application development following the MVC (Model-View-Controller) pattern, which is a widely used model for contemporary web programming.

### Introduction to ReactJs Framework (Front End)

#### Overview of ReactJs Framework

React JS is a JavaScript library used for building user interfaces (UI). React is widely utilized and comes with a diverse and rich ecosystem. While UI is undoubtedly important, it's not everything. To develop a complete application, React JS alone cannot handle everything; you will need to add:

* Server-side language: to handle logic and store data on the server.
* HTML/CSS if you are building a web application.
* Flux/Redux: a architecture that helps you organize code clearly and cleanly.
* Objective C: if you are using React to build an iOS app.

#### Benefits of using ReactJs Framework

Whatever you do, there must be a reason. As more and more users use Reactjs, Reactjs must bring good benefits.

* ReactJS makes writing JavaScript code easier by using a special syntax called JSX. Through JSX, developers can embed HTML and JavaScript code seamlessly.
* ReactJS allows developers to break down complex UI structures into independent components. Developers no longer have to worry about the overall web application; now, they can easily break down complex UI/UX structures into simpler components.
* Accompanying ReactJS are numerous development tools that make code debugging more straightforward.
* Another advantage of ReactJS is its SEO-friendliness. While many JS frameworks are not SEO-friendly, ReactJS, with the support of data rendering on the web page, contributes to a more standardized SEO approach.

### Overview of Mysql database

* MySQL is the world's most popular open-source relational database management system, highly favored by developers during application development. Due to its high speed, stability, user-friendliness, portability, and compatibility across multiple operating systems, MySQL provides a vast array of powerful utility functions.
* With its high speed and robust security features, MySQL is particularly suitable for applications with database access on the internet. Users can download MySQL for free from its official website. MySQL has various versions tailored for different operating systems, including Win32 for Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS, and more.
* MySQL is a fundamental example of a relational database management system using Structured Query Language (SQL). It is widely used to support NodeJs, PHP, Perl, and various other languages, serving as a storage repository for information on websites developed using NodeJs, PHP, Perl, and more.

### Overview of RESTful APIs

REST stands for Representational State Transfer. A RESTful API is a standard used in designing APIs for web applications to manage resources. RESTful is one of the widely adopted API design styles today for enabling communication between various applications (web, mobile, etc.). The key function of REST is to define how to use HTTP methods (such as GET, POST, PUT, DELETE) and format URLs for web applications to manage resources. RESTful does not prescribe the application's code logic and is not constrained by the programming language of the application; any language or framework can be used to design a RESTful API.

### Introducing the MVC (Model - View - Controller) model

#### Overview of MVC

The Model-View-Controller (MVC) framework is an architectural/design pattern that separates an application into three main logical components Model, View, and Controller. Each architectural component is built to handle specific development aspects of an application. It isolates the business logic and presentation layer from each other. It was traditionally used for desktop graphical user interfaces (GUIs). Nowadays, MVC is one of the most frequently used industry-standard web development frameworks to create scalable and extensible projects. It is also used for designing mobile apps.

MVC was created by Trygve Reenskaug. The main goal of this design pattern was to solve the problem of users controlling a large and complex data set by splitting a large application into specific sections that all have their own purpose.

The MVC model consists of the following three components:

* Controller: The controller is the component that enables the interconnection between the views and the model, so it acts as an intermediary. The controller doesn’t have to worry about handling data logic, it just tells the model what to do. It processes all the business logic and incoming requests, manipulates data using the Model component, and interacts with the View to render the final output.
* View: The View component is used for all the UI logic of the application. It generates a user interface for the user. Views are created by the data which is collected by the model component, but these data aren’t taken directly but through the controller. It only interacts with the controller.
* Model: The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. It can add or retrieve data from the database. It responds to the controller’s request because the controller can’t interact with the database by itself. The model interacts with the database and gives the required data back to the controller.

#### Benefits of using the MVC model

* Clear processing sequence.
* Establishes a standardized model for the project, making it more accessible for individuals with expertise outside the project.
* Clearly distinguishes components, facilitating a smoother development, management, operation, and maintenance process, creating specialized functions and enabling control over the processing flow.

# SYSTEM ANALYSIS AND DESIGN

## Analyze project requirements

### Topic Description

This is a fashion business website, combined with BlockChain technology. The website will sell and introduce a variety of products from many brands with clear product details, and have the following typical functions:

* Registration/login options: Website allows users to register for a new account or log in directly with Facebook or Google accounts. At the same time, record customer information, number of login times and summarize general purchasing behaviors, support updating promotional news right on each customer's personal page.
* Decentralized management: The main admin can set internal admin rights, decentralize, add, remove and edit management functions through the Administration page.
* Search: Smart search bar helps customers search for products/articles based on related keywords, supports advanced search and integrates many filters.
* Edit interface: Admin has the right to change the basic colors and fonts of the website.
* Content management: Function to manage adding and removing articles, categories, and editing articles.
* Media management: Admin manages, adds, removes, and edits images and videos
* Online customer care function: Allows customers to interact directly with users through the website. Display a pop-up chat frame that appears every time a customer accesses the website, the chat content leads to the server or the business's customer care department to consult on product information.
* Product attribute options: Add product attributes (color, size, price...). Allow customers to leave comments below each product, rate the product on a scale of 1-5 How to create an objective and professional shopping space. Product categories can be divided into many levels.
* Discount, coupon: Add discount codes, custom promotions.
* Backup & Restore: Backup and Restore data in seconds.
* Admin has the right to create unlimited categories, products, and create digital products (download format).
* Product filter: Can filter products based on many forms (category, brand, attributes...)
* Multi-currency: Enter the exchange rate and the price of all products will change automatically. Integrates many different currencies.
* Payment gateway: Supports many popular payment gateways to create the best purchasing conditions for customers such as COD, ATM, Credit Card, Paypal, BlockChains.
* Recurring payments: Recurring payments can be set up for products.
* Shipping: Allows customers to choose the desired delivery/receipt method. Process orders according to the process of Order Receiving - Confirmation - Package - Shipping - Success. Customers can completely monitor these processes through their personal accounts. Create maximum convenience for users from the access stage until deciding to buy the product.
* Sales report: Provides an overview of sales and daily, monthly, and yearly revenue from comprehensive to detailed information.
* Quick purchase: Ordering feature with just 1 click.
* News: Integrate news page and blog features.
* Gallery: Create photo albums and slideshows on the website.
* Import, export: Import and export products, customers, categories using excel files, easily manage and quickly post many products.
* Compare products: Customers can compare many products on the website.
* Related products: When customers view a product, the system will list similar products below to give customers more choices. Set a mode to remind customers to buy previously viewed products, allowing products to be stored in the shopping cart.
* There are a full range of communication channels with customers: These channels include 3 main groups (Interaction via social network channels - Comments, customer comments on products - Q&A).
* Print invoices: automatically create invoice files in admin, easy to print.
* Warehouse management: easily manage products and quantities, with bulk editing function in admin

### Topic Description

**User request**

Customers are individuals with shopping needs who will search for necessary items within the system and proceed to purchase these items. Therefore, the system needs to have the following functionalities:

* Customers register accounts to receive various benefits.
* Multiple selection methods as well as a variety of product types.
* Display a list of store items for customers to view, select, and purchase.
* Customers can view the latest news and promotions on the website.
* After customers make their selections and place orders, an order summary should appear for customers to enter purchasing information and view the purchase invoice.

**Administrator request**

The owner has the authority to control all activities within the system. This individual is provided with a username and password to log in to the system and perform the following functions:

* Update, add, edit, and delete items, product categories, manufacturers, and other specific information requiring accuracy.
* Receive, verify, and confirm orders placed through the customer website.
* Generate statistics on daily sales revenue, popular items, and conduct monthly revenue analysis for best-selling and slow-moving items.
* Customer care: Receive and respond to customer questions and feedback when necessary. Additionally, provide advice and assistance to customers with needs.
* Manage user accounts and permissions.

**Non-functional requirements**

* Page Loading Speed: The website's page loading speed must ensure promptness at every user experience stage, from when they first access to view detailed product or service information to when they add items to the cart and proceed to the final payment step.
* Versatility: The interface design of the website needs to be visually appealing, user-friendly, and easy to navigate for users in all operations, including searching for products or services, viewing detailed content and images, placing orders, making payments, and tracking the delivery process.
* Security: The security of an e-commerce website is an extremely crucial concern. Businesses need to ensure the establishment of a secure system that protects all customer information and personal data throughout account registration, order placement, and product payment steps to mitigate security risks during operation and comply with information security regulations for customers.
* Integration with Social Media: Customers, after reviewing or successfully confirming orders for products or services on the online platform, often desire to share these experiences with their friends and family on social media. Helps businesses promote their products widely to more customers.
* Online Payment: This capability enhances the convenience of shopping and making payments directly on the business's website, ensuring professionalism and a high level of reliability for our brand.

## Describes the entity's properties in detail

### A diagram of a network with Ice hockey rink in the background Description automatically generatedAddressUser entity

Figure 2. - Address User entity

### A diagram of a computer code Description automatically generatedAllcode entity

Figure 2. - Allcode entity

### A diagram of a company Description automatically generatedBanner entity

Figure 2. - Banner entity

### Blog entity

A diagram with text on it

Description automatically generated

Figure 2. - Blog entity

### Message entity

A diagram of a message

Description automatically generated

Figure 2. - Message entity

### OrderDetail entity

A diagram of a company

Description automatically generated

Figure 2. - OrderDetail entity

### OrderProduct entity

A diagram of a diagram

Description automatically generated

Figure 2. - OrderProduct entity

### ProductDetail entity

A diagram of a product

Description automatically generated

Figure 2. - ProductDetail entity

### ProductDetailSize entity

A diagram of a company

Description automatically generated

Figure 2. - ProductDetailSize entity

### ProductImage entity

A diagram of a product

Description automatically generated

Figure 2. - ProductImage entity

### Product entity

A diagram of a product

Description automatically generated

Figure 2. - Product entity

### RoomMessage entity

A diagram of a room

Description automatically generated

Figure 2. - RoomMessage entity

### A diagram of a product Description automatically generatedShopCart entity

Figure 2. - ShopCart entity

### TypeShip entity

A diagram of types of text

Description automatically generated

Figure 2. - TypeShip entity

### TypeVoucher entity

**A diagram of a type of voucher

Description automatically generated**

Figure 2. - TypeVoucher entity

### User entity

Figure 2. - User entity

### Voucher entity

A diagram of a diagram

Description automatically generated

Figure 2. - Voucher entity

### VoucherUsed entity

A diagram of a voucher used

Description automatically generated

Figure 2. - VoucherUsed entity

### Comment entity

A diagram of a diagram

Description automatically generated

Figure 2. - Comment entity

### Supplier entity

A diagram with text and circles

Description automatically generated with medium confidence

Figure 2. - Supplier entity

### Receipt entity

A diagram of a network

Description automatically generated

Figure 2. - Receipt entity

### ReceiptDetail entity

A diagram of a company

Description automatically generated

Figure 2. - ReceiptDetail entity

## E-R entity relationship symbol

A diagram of a network

Description automatically generated

Figure 2. - E-R entity relationship symbol

## Database schema

A computer screen shot of a computer

Description automatically generated

Figure 2. - Database diagram

## Data tables

### AddressUser table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark289) | int(11) | No |  | Primary key, auto\_increment | Account address code |
| [userId](#_bookmark292) | int(11) | Yes | NULL | Foreign key | Account code |
| [shipName](#_bookmark295) | varchar(255) | Yes | NULL |  | Recipient's name |
| [shipAdress](#_bookmark298) | varchar(255) | Yes | NULL |  | Receiver's address |
| [shipEmail](#_bookmark300) | varchar(255) | Yes | NULL |  | Recipient email address |
| [shipPhonenumb er](#_bookmark302) | varchar(255) | Yes | NULL |  | Recipient phone number |
| [createdAt](#_bookmark304) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark306) | datetime | No |  |  | Edit date |

Table 2. - AddressUser table

### Allcode table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark180) | int(11) | No |  | Primary key, auto\_increment | Identifier |
| [type](#_bookmark181) | varchar(255) | Yes | NULL |  | Code type |
| [value](#_bookmark182) | varchar(255) | Yes | NULL |  | Value |
| [code](#_bookmark183) | varchar(255) | Yes | NULL | Unique | Code |
| [createdAt](#_bookmark184) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark185) | datetime | No |  |  | Edit date |

Table 2. - Allcode table

### Banner table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark269) | int(11) | No |  | Primary key, auto\_increment | Banner code |
| [description](#_bookmark271) | longtext | Yes | NULL |  | Describe |
| [name](#_bookmark273) | varchar(255) | Yes | NULL |  | Banner name |
| [statusId](#_bookmark275) | varchar(255) | Yes | NULL | Foreign Key | Status code |
| [image](#_bookmark277) | longblob | Yes | NULL |  | Image |
| [createdAt](#_bookmark279) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark280) | datetime | No |  |  | Edit date |

Table 2. - Banner table

### Blog table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark188) | int(11) | No |  | Primary key, auto\_increment | Post code |
| [shortdescription](#_bookmark189) | longtext | Yes | NULL |  | Short description |
|  |  |  |  |  |  |
| [title](#_bookmark190) | varchar(255) | Yes | NULL |  | Title |
| [subjectId](#_bookmark191) | varchar(255) | Yes | NULL |  | Theme code |
| [statusId](#_bookmark192) | varchar(255) | Yes | NULL |  | Status code |
| [image](#_bookmark193) | longblob | Yes | NULL |  | Image |
| [contentMarkdown](#_bookmark194) | longtext | Yes | NULL |  | Markdown content in text format |
| [contentHTML](#_bookmark195) | longtext | Yes | NULL |  | Markdown content in html format |
| [userId](#_bookmark196) | int(11) | Yes | NULL | Foreign Key | User code |
| [view](#_bookmark197) | int(11) | Yes | NULL |  | View |
| [createdAt](#_bookmark198) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark199) | datetime | No |  |  | Edit date |

Table 2. - Blog table

### Comment table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark289) | int(11) | No |  | Primary key, auto\_increment | Comment/review code |
| [content](#_bookmark292) | longtext | Yes | NULL |  | Content |
| [image](#_bookmark295) | longblob | Yes | NULL |  | Image |
| [parentId](#_bookmark298) | int(11) | Yes | NULL |  | Comment/review code |
| [productId](#_bookmark300) | int(11) | Yes | NULL | Foreign Key | Product code |
| [userId](#_bookmark302) | int(11) | Yes | NULL | Foreign Key | User code |
| [blogId](#_bookmark304) | int(11) | Yes | NULL | Foreign Key | Post code |
| [star](#_bookmark306) | int(11) | Yes | NULL |  | Number of stars |
| [createdAt](#_bookmark307) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark308) | datetime | No |  |  | Edit date |

Table 2. - Comment table

### Message table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark201) | int(11) | No |  | Primary key, auto\_increment | Message code |
| [text](#_bookmark202) | longtext | Yes | NULL |  | Content |
| [userId](#_bookmark203) | int(11) | Yes | NULL | Foreign Key | User code |
| [roomId](#_bookmark204) | int(11) | Yes | NULL | Foreign Key | Room code |
| [unRead](#_bookmark205) | tinyint(1) | Yes | NULL |  | Message reading status |
| [createdAt](#_bookmark206) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark207) | datetime | No |  |  | Edit date |

Table 2. - Message table

### OrderDetail table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark290) | int(11) | No |  | Primary key, auto\_increment | Order detail code |
| [orderId](#_bookmark293) | int(11) | Yes | NULL | Foreign Key | Code orders |
| [productId](#_bookmark296) | int(11) | Yes | NULL | Foreign Key | Product size code |
| [quantity](#_bookmark299) | int(11) | Yes | NULL |  | Quantity |
| [realPrice](#_bookmark301) | bigint(20) | Yes | NULL |  | Real price |
| [createdAt](#_bookmark303) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark305) | datetime | No |  |  | Edit date |

Table 2. - OrderDetail table

### OrderProduct table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark209) | int(11) | No |  | Primary key, auto\_increment | Code orders |
| [addressUserId](#_bookmark211) | int(11) | Yes | NULL | Foreign Key | User address code |
| [statusId](#_bookmark212) | varchar(255) | Yes | NULL | Foreign Key | Status code |
| [typeShipId](#_bookmark213) | int(11) | Yes | NULL | Foreign Key | Shipping type code |
| [voucherId](#_bookmark214) | int(11) | Yes | NULL | Foreign Key | Promotion code |
| [note](#_bookmark215) | varchar(255) | Yes | NULL |  | Note |
| [isPaymentOnlie n](#_bookmark216) | int(11) | Yes | NULL |  | Payment status |
| image | longblob | Yes | NULL |  | Image |
| [createdAt](#_bookmark217) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark218) | datetime | No |  |  | Edit date |

Table 2. - OrderProduct table

### ProductDetail table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark220) | int(11) | No |  | Primary key, auto\_increment | Product type code |
| [productId](#_bookmark221) | int(11) | Yes | NULL | Foreign Key | Product code |
| [description](#_bookmark222) | longtext | Yes | NULL |  | Describe |
| [nameDetail](#_bookmark223) | varchar(255) | Yes | NULL |  | Product type name |
| [originalPrice](#_bookmark224) | bigint(20) | Yes | NULL |  | Cost |
| [discountPrice](#_bookmark225) | bigint(20) | Yes | NULL |  | Promotional price |
| [createdAt](#_bookmark226) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark227) | datetime | No |  |  | Edit date |

Table 2. - ProductDetail table

### ProductDetailSize table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark220) | int(11) | No |  | Primary key, auto\_increment | Product type size code |
| [productdetailId](#_bookmark221) | int(11) | Yes | NULL | Foreign Key | Product type code |
| [width](#_bookmark222) | varchar(255) | Yes | NULL |  | Width |
| [height](#_bookmark223) | varchar(255) | Yes | NULL |  | Height |
| [weight](#_bookmark224) | varchar(255) | Yes | NULL |  | Weight |
| [sizeId](#_bookmark225) | varchar(255) | Yes | NULL | Foreign Key | Size code |
| [createdAt](#_bookmark227) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark228) | datetime | No |  |  | Edit date |

Table 2. - ProductDetailSize table

### ProductImage table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark285) | int(11) | No |  | Primary key, auto\_increment | Product type image code |
| [caption](#_bookmark286) | varchar(255) | Yes | NULL |  | Describe |
| [productdetailId](#_bookmark287) | int(11) | Yes | NULL | Foreign Key | Product type code |
| [image](#_bookmark291) | longblob | Yes | NULL |  | Image |
| [createdAt](#_bookmark294) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark297) | datetime | No |  |  | Edit date |

Table 2. - ProductImage table

### Product table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark180) | int(11) | No |  | Primary key, auto\_increment | Product code |
| [name](#_bookmark181) | varchar(255) | Yes | NULL |  | Product's name |
| [contentHTML](#_bookmark182) | longtext | Yes | NULL |  | Markdown content in html format |
| [contentMarkdo wn](#_bookmark183) | longtext | Yes | NULL |  | Markdown content in text format |
| [statusId](#_bookmark184) | varchar(255) | Yes | NULL | Foreign Key | Status code |
| [categoryId](#_bookmark185) | varchar(255) | Yes | NULL | Foreign Key | Category code |
| [view](#_bookmark229) | int(11) | Yes | NULL |  | View |
| [madeby](#_bookmark230) | varchar(255) | Yes | NULL |  | Made/designed by |
| [material](#_bookmark231) | varchar(255) | Yes | NULL |  | Material |
| [brandId](#_bookmark232) | varchar(255) | Yes | NULL | Foreign Key | Brand code |
| [createdAt](#_bookmark233) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark234) | datetime | No |  |  | Edit date |

Table 2. - Product table

### RoomMessage table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark236) | int(11) | No |  | Primary key, auto\_increment | Room code |
| [userOne](#_bookmark237) | int(11) | Yes | NULL | Foreign Key | First participant code |
| [userTwo](#_bookmark238) | int(11) | Yes | NULL | Foreign Key | Second participant code |
| [createdAt](#_bookmark239) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark240) | datetime | No |  |  | Edit date |

Table 2. - RoomMessage table

### ShopCart table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark242) | int(11) | No |  | Primary key, auto\_increment | Cart code |
| [userId](#_bookmark243) | int(11) | Yes | NULL | Foreign Key | User code |
| productdetailsizeId | int(11) | Yes | NULL | Foreign Key | Product type size code |
| [quantity](#_bookmark272) | int(11) | Yes | NULL |  | Quantity |
| [statusId](#_bookmark274) | varchar(255) | Yes | NULL |  | Cart status |
| [createdAt](#_bookmark276) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark278) | datetime | No |  |  | Edit date |

Table 2. - ShopCart table

### TypeShip table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark201) | int(11) | No |  | Primary key, auto\_increment | Shipping type code |
| [type](#_bookmark202) | varchar(255) | Yes | NULL |  | Type name |
| [price](#_bookmark203) | bigint(20) | Yes | NULL |  | Giá tiền |
| [createdAt](#_bookmark204) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark205) | datetime | No |  |  | Edit date |

Table 2. - TypeShip table

### TypeVoucher table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark245) | int(11) | No |  | Primary key, auto\_increment | Promotion type code |
| [typeVoucher](#_bookmark246) | varchar(255) | Yes | NULL |  | Type name |
| [value](#_bookmark247) | bigint(20) | Yes | NULL |  | Value |
| [maxValue](#_bookmark248) | bigint(20) | Yes | NULL |  | Maximum value |
| [minValue](#_bookmark249) | bigint(20) | Yes | NULL |  | Minimum value |
| [createdAt](#_bookmark250) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark251) | datetime | No |  |  | Edit date |

Table 2. - TypeVoucher table

### User table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark310) | int(11) | No |  | Primary key, auto\_increment | User code |
| [email](#_bookmark311) | varchar(255) | Yes | NULL |  | Email |
| [password](#_bookmark312) | varchar(255) | Yes | NULL |  | Password |
| [firstName](#_bookmark313) | varchar(255) | Yes | NULL |  | Surname |
| [lastName](#_bookmark314) | varchar(255) | Yes | NULL |  | Name |
| [address](#_bookmark315) | varchar(255) | Yes | NULL |  | Address |
| [genderId](#_bookmark316) | varchar(255) | Yes | NULL | Foreign Key | Gender code |
| [phonenumber](#_bookmark317) | varchar(255) | Yes | NULL |  | Phone number |
| [image](#_bookmark318) | longblob | Yes | NULL |  | Image |
| [dob](#_bookmark319) | varchar(255) | Yes | NULL |  | Date of birth |
| [roleId](#_bookmark320) | varchar(255) | Yes | NULL | Foreign Key | Role code |
| [statusId](#_bookmark321) | varchar(255) | Yes | NULL | Foreign Key | Status code |
| [isActiveEmail](#_bookmark322) | tinyint(1) | Yes | NULL |  | Email activation status |
| [usertoken](#_bookmark323) | varchar(255) | Yes | NULL |  | User tokens |
| [createdAt](#_bookmark324) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark325) | datetime | No |  |  | Edit date |

Table 2. - User table

### Voucher table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark253) | int(11) | No |  | Primary key, auto\_increment | Promotional code |
| [fromDate](#_bookmark254) | varchar(255) | Yes | NULL |  | Start day |
| [toDate](#_bookmark255) | varchar(255) | Yes | NULL |  | End date |
| [typeVoucherId](#_bookmark256) | int(11) | Yes | NULL | Foreign Key | Promotion type code |
| [amount](#_bookmark257) | int(11) | Yes | NULL |  | Quantity |
| [codeVoucher](#_bookmark259) | varchar(255) | Yes | NULL |  | Promotional code |
| [createdAt](#_bookmark260) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark261) | datetime | No |  |  | Edit date |

Table 2. - Voucher table

### VoucherUser table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark263) | int(11) | No |  | Primary key, auto\_increment | ID code |
| [voucherId](#_bookmark264) | int(11) | No |  | Foreign Key | Voucher code |
| [userId](#_bookmark265) | int(11) | No |  | Foreign Key | User code |
| [status](#_bookmark266) | int(11) | No | 0 |  | Status |
| [createdAt](#_bookmark267) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark268) | datetime | No |  |  | Edit date |

Table 2. - VoucherUser table

### Supplier table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark343) | int(11) | No |  | Primary key, auto\_increment | Supplier code |
| [name](#_bookmark344) | varchar(255) | Yes | NULL |  | Supplier Name |
| [address](#_bookmark345) | varchar(255) | Yes | NULL |  | Address |
| [phonenumber](#_bookmark346) | varchar(255) | Yes | NULL |  | Phone number |
| [email](#_bookmark347) | varchar(255) | Yes | NULL |  | Email |
| [createdAt](#_bookmark364) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark365) | datetime | No |  |  | Edit date |

Table 2. - Supplier table

### Receipt table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark295) | int(11) | No |  | Primary key, auto\_increment | Import code |
| [userId](#_bookmark296) | int(11) | Yes | NULL | Foreign Key | Employee code |
| [supplierId](#_bookmark302) | int(11) | Yes | NULL | Foreign Key | Supplier code |
| [createdAt](#_bookmark303) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark304) | datetime | No |  |  | Edit date |

Table 2. - Receipt table

### ReceiptDetail table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Constraint** | **Description** |
| [id](#_bookmark270) | int(11) | No |  | Primary key, auto\_increment | Detailed import code |
| [receiptId](#_bookmark271) | int(11) | Yes | NULL | Foreign Key | Import code |
| [productDetailSi zeId](#_bookmark272) | int(11) | Yes | NULL | Foreign Key | Product size code |
| [quantity](#_bookmark273) | int(11) | Yes | NULL |  | Quantity |
| [price](#_bookmark274) | bigint(20) | Yes | NULL |  | Unit price |
| [createdAt](#_bookmark275) | datetime | No |  |  | Date created |
| [updatedAt](#_bookmark276) | datetime | No |  |  | Edit date |

Table 2. - ReceiptDetail table

## Description: Diagram Description automatically generatedActors in the request

Figure 2. - Use Case Admin

Based on the specified requirements, we have the following actors:

**Administrator - Admin:** The individual with the highest responsibility for the system, possessing all system rights, including account management, category management, product management, order management, brand management, post management, banner management, promotion management, message management, shipping type management, pagination management, statistics management, and more.

Description: Diagram  Description automatically generated with medium confidence**Member (registered user - User):** is a regular user who has registered an account with the system and has the right to view products, posts, manage shopping carts, and manage personal information.

Figure 2. - Use Case User

Salesperson: is a salesperson who has registered an account and is in charge of sales, tracking order information and importing goods.

**

Figure 2. - Usercase Saler

Description: Diagram  Description automatically generatedVisitor (unregistered user - Viewer): is a user who has not registered or not logged into the website. They only have the right to view products, posts, and register a new account.

Figure 2. - Use Case Viewer

## Use Cases in the system

### General Use cases

A diagram of a company

Description automatically generated

Figure 2. - General use cases

### Use Case of admin

User management function: allowing administrators to add, edit, search, delete user accounts, view user information, and export to excel.

Figure 2. - Use Case User management

Product management function: allows administrators to add, edit, search, delete products, view detailed product information, product types, export to Excel.

Figure 2. - Use Case Product Management

Brand management function: allows administrators to add, edit, search, delete product brands, view detailed information, and export to excel.

Figure 2. - Use Case Brand management

Post management function: allows administrators to add, edit, search, delete news posts, view detailed information, export to excel.

Figure 2. 33 - Use Case Manage posts

Promotion management function: allows administrators to add, edit, search, delete promotion codes, view detailed information, export to excel

Figure 2. 34 - Use Case Promotion management

Promotion type management function: allowing administrators to add, edit, search, delete promotion types, view detailed information, and export to excel.

Figure 2. 35 - Use Case Manage promotion types

Shipping type management function: allowing administrators to add, edit, search, delete shipping types, view detailed information, and export to Excel.

Figure 2. 36 - Use Case Manage shipping types

Order management function: allowing administrators to search, view detailed information, update order status, and export to Excel.

Figure 2. 37 - Use Case Order management

Topic management function: allows administrators to add, edit, search, delete post topics, view detailed information, and export to Excel.

Figure 2. - Use case Topic management

Banner management function: allows administrators to add, edit, search, delete advertising banners, view detailed information, and export to Excel.

Figure 2. 39 - Use case Banner management

Message management function, allowing administrators to text messages with registered users.

Figure 2. 40 - Use case Manage messages

Supplier management function: allows administrators to add, edit, search, delete suppliers, view detailed information, and export to Excel.

Figure 2. 41 - Use case Supplier management

Goods import management function: allowing administrators to add, edit, search, delete goods orders, view detailed information, and export to Excel.

Figure 2. 42 - Use case Management of import goods

Statistics function, allowing statistical management

* Total number of orders, products, reviews, users.
* Order status rate chart
* Revenue chart by month of the year
* Revenue chart by day of the month
* List of profits by day, month, year

Figure 2. 43 - Use case Statistical

### Use Case of user

Personal information management function: allowing members to edit and view detailed personal information.

Figure 2. - Use case Manage personal information

Personal order management function: allows members to review information about orders placed in the system and change status.

Figure 2. - Use case Manage personal orders

Personal address management function: allowing members to add, edit, and delete personal addresses.

Figure 2. - Use case Manage personal addresses

Personal promotion management function: allows members to review promotion codes saved on the system.

Figure 2. - Use case Manage personal promotions

The add product to cart function: allows members to select products and add to cart.

Figure 2. - Use case Add product

The account registration function: allows members to open an account on the system to log in and place orders.

Figure 2. - Use case Register an account

Ordering and payment function: allows members to create orders and choose online or cash payment methods.

Figure 2. - Use case Oder payment

Comment function: rating for members to comment on posts and rate products.

Figure 2. - Use case Comments, reviews

Product viewing function: allows members to view all product information in the system.

Figure 2. 52 - Use case View products

News post viewing function: allows members to view all information of fashion news posts in the system.

Figure 2. 53 - Use case View news posts

### Use Case of a visitor

Product viewing function: allows visitors to view all product information in the system.

Figure 2. 54 - Use case View products

The news post viewing function: allows visitors to view all information of fashion news posts in the system.

Figure 2. 55 - Use case View news posts

The account registration function: allows visitors to open an account on the system to log in and place orders.

Figure 2. 56 - Use case Register

### Use case of a salesperson

The account login function allows sales staff to log in to the admin page.

Figure 2. 57 - Use case Login

Order management function: allowing sales staff to search, view detailed information, update order status, and export to Excel.

Figure 2. 58 - Use case Order management

Supplier management function: allowing employees to add, edit, search, delete suppliers, view detailed information, and export to Excel.

Figure 2. 59 - Use case Supplier management

Goods import management function: allowing administrators to add, edit, search, delete goods orders, view detailed information, and export to Excel.

Figure 2. 60 - Use case Management of import goods

Message management function, allowing employees to message users with accounts.

Figure 2. 61 - Use case Messages

## List of actors

|  |  |  |
| --- | --- | --- |
| **STT** | **Agent** | **Describe** |
| 1 | Admin | The individual with the highest responsibility for the system, having all system rights, including user management, category management, message management, banner management, brand management, theme management, post management, product management, promotion management, discount type management, shipping type management, order management, statistics management, supplier management, and inventory management. |
| 2 | Member | Regular users, registered with the system, have the privilege to view order history, manage personal information, add products to the shopping cart, manage personal addresses, comment on posts, review products, register new accounts, manage promotions, place orders for payment, register accounts, view product lists, and view post lists. |
| 3 | Visitor | Users who have not registered or logged in to the website only have the right to view product information, post information, and register an account. |
| 5 | Sales agent | Sales staff have an account opened by the admin to log into the admin page and have the right to manage orders, manage messages, manage suppliers, and manage goods imports. |

Table 2. - List of actors

## List of Use cases

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Agent name** | **Function name** | **Describe** |
| 1 | Admin | User management | User management function, allowing administrators to search, search for users, and view user information on the system. |
| Category management | The all-category management function allows administrators to search, add, edit, delete, and view all categories in the system. |
| Banner management | The all-banner management function allows administrators to search, add, edit, delete, and view all banners in the system. |
| Statistical | The revenue and profit statistics function allows administrators to view the total revenue and total profit of all completed orders. |
| Manage messages | The message management function allows administrators to message users on the system. |
| Brand management | The all brand management function allows administrators to search, add, edit, delete, and view all brands in the system. |
| Topic management | The all topic management function allows administrators to search, add, edit, delete, and view all topics in the system. |
| Manage posts | All posts management function allows administrators to search, add, edit, delete, and view all posts in the system. |
| Product Management | The all product management function allows administrators to search, add, edit, delete, and view all products in the system. |
| Promotion management | The all promotions management function allows administrators to search, add, edit, delete, and view all promotions in the system. |
| Manage promotion types | The function of managing all types of promotions allows administrators to search, add, edit, delete, and view all types of promotions in the system. |
| Manage ship types | The function of managing all types of ships allows administrators to search, add, edit, delete, and view all types of ships in the system. |
| Order management | The all order management function allows administrators to search, view details, and update the status of all orders in the system. |
| Supplier management | The all supplier management function allows administrators to search, add, edit, delete, and view all suppliers in the system. |
| Management of import goods | The function of managing all import orders allows administrators to search, create purchase orders, and view all import orders in the system. |
| 2 | User | Manage personal information | The personal information management function allows members to edit and view their personal information in the system. |
| Manage personal orders | The personal order management function allows members to review information about orders placed in the system and change status. |
| Manage personal addresses | The personal address management function allows members to add, edit, delete, and view all of their personal addresses. |
| Add product to cart | The function of adding products to the cart allows members to add products, update quantities, and delete products in the system's cart. |
| Promotion | Promotion function allows members to save promotion codes and review all saved codes. |
| Order payment | The payment order function allows members to order and pay by cash or online. |
| Comments and reviews | The review comment function allows members to comment on posts and rate products. |
| Register an account | The account registration function allows members to create personal accounts and log in to the system. |
| 3 | Viewer | View product information | The product information viewing function allows visitors to view full details of each product in the system. |
| View post information | The post information viewing function allows visitors to view full details of each post in the system. |
| Register an account | The account registration function allows visitors to create personal accounts and log in to the system. |
| **4** | Sales agent | Log in | The account login function allows shippers to log in to the website's mobile application. |
| Order management | Order management function, allowing sales staff to search, view detailed information, update order status, and export to Excel. |
| Supplier management | Supplier management function, allowing employees to add, edit, search, delete suppliers, view detailed information, and export to Excel. |
| Manage messages | Message management function, allowing employees to message users with accounts |
| Management of import goods | Goods management function, allowing administrators to add, edit, search, delete orders, view detailed information, and export to Excel |

Table 2. - List of Use cases

## Use Case diagram based on each actor detail

A diagram of a person's work flow

Description automatically generated

Figure 2. - Use case diagram according to the actor is Admin

A diagram of a person's work flow

Description automatically generated

Figure 2. - Use case diagram for each User actor

A diagram of a diagram

Description automatically generated

Figure 2. - Use case diagram for each guest actor

A diagram of a company

Description automatically generated

Figure 2. - Usecase diagram for each salesperson

## Specification of Use Cases

### Use case for visiting guests

***Use case specification to view product information:***

* Actor: viewer, visitor
* Pre-conditions: the internet works normally
* Post-condition: user does not have a login account or has not logged into the website
* Trigger: viewer, visitor
* Event stream:
* Access the website address
* Select the store section, click on the product
* The website displays full detailed information of each product and product list...

***Use case specification see post information:***

* Actor: viewer, visitor
* Pre-conditions: the internet works normally
* Post-condition: user does not have a login account or has not logged into the website
* Trigger: viewer, visitor
* Event stream:
* Access the website address
* Select the news section, click on each post
* The website displays full detailed information for each post and list of posts

***Account registration Use case specification:***

* Actor: viewer, visitor
* Pre-conditions: the internet works normally
* Post-condition: user does not have a login account or has not logged into the website
* Trigger: viewer, visitor
* Event stream:
* Access the website address
* Select registration
* Fill full the information
* Click register to create an account

### Use case for visiting guests

***Use case specification for personal information management:***

* Actor: user, member
* Pre-conditions: the internet works normally
* Post-conditions: members must have an account and be able to log in to the website
* Activation agent: user, member
* Event stream:
* Log in to the website
* Select personal information and profile section
* Website displays personal information
* View and edit data (if necessary)
* Click update
* Notify success or error messages.

***Use case specification for personal order management:***

* Actor: user, member
* Pre-conditions: the internet works normally
* Post-conditions: members must have an account and be able to log in to the website
* Activation agent: user, member
* Event stream:
* Log in to the website
* Select personal information, purchase order
* The website displays information about orders that members have placed in the past
* Click on the cancel order or received order button to change the order status

***Use case specification for personal address management:***

* Actor: user, member
* Pre-conditions: the internet works normally
* Post-conditions: members must have an account and be able to log in to the website
* Activation agent: user, member
* Event stream:
* Access the website address
* Select personal information and address
* View and create new addresses, or edit and delete as needed.

***Use case specification of adding product to cart:***

* Actor: user, member
* Pre-conditions: the internet works normally
* Post-conditions: members must have an account and be able to log in to the website
* Activation agent: user, member
* Event stream:
* Access the website address
* Add product to cart
* Update quantity

***Account registration Use case specification:***

* Actor: user, member
* Pre-conditions: the internet works normally
* Post-condition: user does not have a login account or has not logged into the website
* Trigger: viewer, visitor
* Event stream:
* Access the website address
* Select registration
* Fill full the information
* Click register to create an account

***Order and payment Use case specification:***

* Actor: user, member
* Pre-conditions: the internet works normally
* Post-conditions: members must have an account and be able to log in to the website
* Activation agent: user, member
* Event stream:
* Access the website address
* Select cart item
* Select delivery address
* Select promotional code
* Select shipping unit
* Choose online payment method, cash or MetaMask wallet

***Use case specifications and comments:***

* Actor: user, member
* Pre-conditions: the internet works normally
* Post-conditions: members must have an account and be able to log in to the website
* Activation agent: user, member
* Event stream:
* Access the website address
* Select the news section, select a post, comment on the post
* Select the store section, select 1 product, go to the review section and post a product review

### Use case Admin

***User management Use case specification:***

* Actor: admin, manager
* Pre-conditions: the internet works normally
* Post-conditions: admin must be granted management rights and must be able to log in to the administration system
* Activating agent: admin, manager
* Event stream:
* Log in to the administration system
* Select user management
* The system displays a list of data
* Search data

After finding the desired data, the admin has the right to add, edit, and delete accounts

Return to home page

***Use case specification manages all categories:***

* Actor: admin, manager
* Pre-conditions: the internet works normally
* Post-conditions: admin must be granted management rights and must be able to log in to the administration system
* Activating agent: admin, manager
* Event stream:
* Log in to the administration system
* Select the categories to work on
* Add, edit, delete, search data
* Click save, update, delete, search to save data (depending on usage conditions)

***Use case specification for revenue and profit statistics:***

* Actor: admin, manager
* Pre-conditions: the internet works normally
* Post-conditions: admin must be granted management rights and must be able to log in to the administration system
* Activating agent: admin, manager
* Event stream:
* Log in to the administration system
* Choose your home page
* Select the statistics section
* Displays the statistics screen

***Order management Use case specification:***

* Actor: admin, manager
* Pre-conditions: the internet works normally
* Post-conditions: admin must be granted management rights and must be able to log in to the administration system
* Activating agent: admin, manager
* Event stream:
* Log in to the administration system
* Select order management
* Add, edit, delete, search for orders
* Click save, update, delete, search and check to save the order (depending on usage conditions)

***Product import management Use case specification:***

* Actor: admin, manager
* Pre-conditions: the internet works normally
* Post-conditions: admin must be granted management rights and must be able to log in to the administration system
* Activating agent: admin, manager
* Event stream:
* Log in to the administration system
* Select item import management
* Add, edit, delete, search for import orders
* Click save, update, search and check to save the import order (depending on usage conditions)

***Message management Use case specification:***

* Actor: admin, manager
* Pre-conditions: the internet works normally
* Post-conditions: admin must be granted management rights and must be able to log in to the administration system
* Activating agent: admin, manager
* Event stream:
* Log in to the administration system
* Select Manage reminders
* Search conversations by name
* Click on the conversation and compose a message

***Use case specification for revenue and profit statistics:***

* Actor: admin, manager
* Pre-conditions: the internet works normally
* Post-conditions: admin must be granted management rights and must be able to log in to the administration system
* Activating agent: admin, manager
* Event stream:
* Log in to the administration system
* Choose your home page
* Select the statistics section
* Displays the statistics screen

### Use case Saler

***Order management Use case specification:***

* Actor: saler, sale staff
* Pre-conditions: the internet works normally
* Post-conditions: sales staff must be granted management rights and must be able to log in to the administration system
* Activating agent: saler, sale staff
* Event stream:
* Log in to the administration system
* Select order management
* Add, edit, delete, search for orders
* Click save, update, delete, search and check to save the order (depending on usage conditions)

***Product import management Use case specification:***

* Actor: saler, sale staff
* Pre-conditions: the internet works normally
* Post-conditions: sales must be granted management rights and must be able to log in to the administration system
* Activating agent: saler, sale staff
* Event stream:
* Log in to the administration system
* Select item import management
* Add, edit, delete, search for import orders
* Click save, update, search and check to save the import order (depending on usage conditions)

***Supplier management Use case specification:***

* Actor: saler, sale staff
* Pre-conditions: the internet works normally
* Post-conditions: sales must be granted management rights and must be able to log in to the administration system
* Activating agent: saler, sale staff
* Event stream:
* Log in to the administration system
* Select supplier management
* Add, edit, delete, search for suppliers
* Click save, update, search to save supplier information (depending on usage conditions)

***Message management Use case specification:***

* Actor: saler, sale staff
* Pre-conditions: the internet works normally
* Post-conditions: saleser must be granted management rights and must be able to log in to the administration system
* Activating agent: saler, sales staff
* Event stream:
* Log in to the administration system
* Select Manage reminders
* Search conversations by name
* Click on the conversation and compose a message

## Activity diagram

### Admin activity diagram

A diagram of a login

Description automatically generated

Figure 2. 66 - Diagram of admin login activities

A diagram of a computer

Description automatically generated

Figure 2. 67 - User management activity diagram

A diagram of a data flow

Description automatically generated

Figure 2. 68 - Activity diagram for managing all categories

A diagram of a diagram

Description automatically generated with medium confidence

Figure 2. 69 - Statistical activity diagram

### User activity diagram

A diagram of a process

Description automatically generated

Figure 2. 70 - Diagram of personal information management activities

A diagram of a computer

Description automatically generated with medium confidence

Figure 2. 71 - Diagram of personal order management activities

A diagram of a product

Description automatically generated

Figure 2. 72 - Activity diagram of adding products to cart

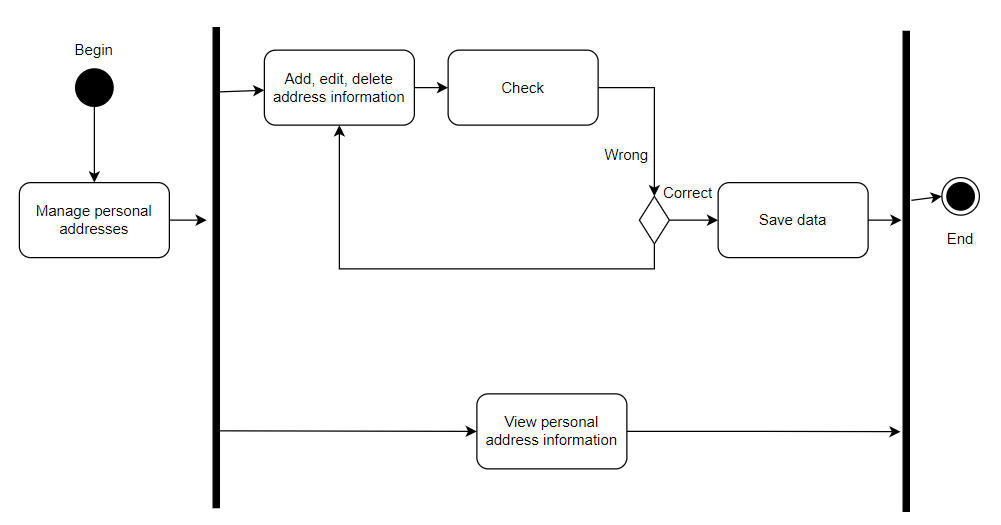


Figure 2. 73 - Personal address management diagram

### Viewer activity diagram

A diagram of a process

Description automatically generated

Figure 2. 74 - Diagram of account registration activities

A white rectangular object with black text

Description automatically generated

Figure 2. 75 - Activity diagram for viewing product information

A close-up of a white rectangular object

Description automatically generated

Figure 2. 76 - Activity diagram for viewing post information

### Activity diagram of sales staff

A screenshot of a diagram

Description automatically generated

Figure 2. 77 - Message management activity diagram

A diagram of a process

Description automatically generated

Figure 2. 78 - Diagram of import management activities

## Algorithm diagram

### Login

A diagram of a computer program

Description automatically generated

Figure 2. 79 - Login flow diagram

### Log out

A diagram of a process

Description automatically generated

Figure 2. 80 - Logout flow diagram

### Delete data

A diagram of a data flow

Description automatically generated

Figure 2. 81 - Data deletion flow diagram

### Updating data

A diagram of a process

Description automatically generated

Figure 2. 82 - Data update flow diagram

### Add data

A diagram of a software system

Description automatically generated

Figure 2. 83 - Flow diagram for adding data

### Cart

A diagram of a computer process

Description automatically generated

Figure 2. 84 - Flow diagram of adding products to cart

# SETUP AND TEST RESULTS

## Software requirements

Program installed on the server. The necessary software for the software to function are:

* Xampp
* Visual studio code
* MetaMask Wallet

To ensure objectivity in comparisons, all programs are tested on the same environment, which is Windows 10/11 with an Intel Core i5-11400 processor and 16GB DDR RAM.

## Design the theme

### List of interfaces

|  |  |  |
| --- | --- | --- |
| **TT** | **Interface name** | **Describe** |
| 1 | Login page | The page helps customers and admins log into the system |
| 2 | Registration page | The page helps customers register for an account |
| 3 | Home page | The first page that opens when a customer visits the website |
| 4 | Store page | The page displays the store's product list |
| 5 | News sites | The page displays a list of the store's fashion posts |
| 6 | Product detail page | The page helps customers view product information |
| 7 | Post details page | The page helps customers view post information |
| 8 | Discount code page | The page helps customers see what discount codes are available and save the codes to their account |
| 9 | Cart page | The page helps customers see the products they have added to their cart |
| 10 | Payment page | The page allows customers to view order information, take notes, and select items. |
| 11 | Profile page | The page allows customers and administrators to view and edit personal information |
| 12 | Address page | The page helps customers add, edit, and delete personal address information |
| 13 | Password change page | The page helps customers and administrators change passwords |
| 14 | Purchase order page | The page helps customers view order history and update order status |
| 15 | Voucher page | The page helps customers review saved voucher codes |
| 16 | Admin home page | The page helps admins view statistics in chart and card form |
| 17 | User management admin page | The page helps admins view, add, update, and delete users |
| 18 | Admin page for category management | The page helps admins view, add, update, and delete categories |
| 19 | Brand management admin page | The page helps admins view, add, update, and delete brands |
| 20 | Product management admin page | The page helps admins view, add, update, and delete products |
| 21 | Vendor management admin page | The page helps admins view, add, update, and delete suppliers |
| 22 | Admin page for import management | The page helps admins view, add, update, and delete goods |
| 23 | Admin page manages banners | The page helps admins view, add, update, and delete banners |
| 24 | The admin page manages topics | The page helps admins view, add, update, and delete topics |
| 25 | The admin page manages posts | The page helps admins view, add, update, and delete posts |
| 26 | The admin page manages ship types | The page helps admins view, add, update, and delete ship types |
| 27 | The admin page manages the voucher house | The page helps admins view, add, update, and delete voucher houses |
| 28 | Order management admin page | The page helps admins view and update orders |
| 29 | The admin page manages messages | The page helps admins view and send messages |
| 30 | Statistics admin page | The page helps admins view revenue and profit statistics by day, month, and year |

Table 3. - Interface list table

### Actual results

## A person holding a purse Description automatically generated

Figure 3. - Home page

Figure 3.1 Home page: The homepage is the first page that users or visitors encounter when accessing the website. At the top, there is a navigation bar leading to other pages. On the upper right side, there are options for logging in and connecting with the MetaMask wallet, as well as links to the shopping cart and personal information page.

A screenshot of a website

Description automatically generated

Figure 3. - Product portfolio page

Figure 3.2 Product portfolio page: The product category page is where the website displays its products. At the top, there is a search bar for product search, filtering options based on available conditions, and a display frame. On the left side are various categories, allowing users to easily search for their desired products through the available categories.

A screenshot of a computer

Description automatically generated

Figure 3. - News page

Figure 3.3 News page: The news page is where various news related to products, fashion, and daily life are featured with the purpose of keeping users informed. On the right side, there is a search bar for searching related news, and below that search bar is a category section with different types of news. This page is designed to facilitate users in easily exploring and staying updated on relevant news.

A screenshot of a website

Description automatically generated

Figure 3. - Voucher page

Figure 3.4 Voucher page: The voucher page is where users can search for product discount codes, free shipping codes, and other relevant discount codes that the website has released for users. These codes can be used to reduce associated fees.

A black sweatshirt with white text

Description automatically generated

Figure 3. - Product details page

Figure 3.5 Product details page: The product detail page is where detailed information about a product is displayed, including the product name, price, category, product status, dimensions (length, width), and available product images.

A screenshot of a computer

Description automatically generated

Figure 3. - Cart page

Figure 3.6 Cart page: The shopping cart page is where users can track the products they have previously added to their cart. Here, details of the products such as price, quantity, and total product cost are displayed. Users also have the option to remove items from the cart. Below that section, there are available shipping methods such as express, standard, and fast. Furthermore, users can enter a voucher code to reduce associated costs.

A screenshot of a computer

Description automatically generated

Figure 3. - Checkout page

Figure 3.7 Checkout page: The checkout page is where users pay for the products they have added to their cart along with their details. Here, users can make payments for selected products. Next to each product will display the price, quantity and total amount. Below that, there are shipping options available such as express, standard, and expedited shipping. Further, users can choose their preferred payment method.

A screenshot of a computer

Description automatically generated

Figure 3. - Payment page using VnPay

Figure 3.8 Payment page using VnPay: The VnPay checkout page is where users can make payments using the VnPay application. On this page, payment information is displayed, and users input relevant details such as the type of goods, amount, payment description, bank, and language. After providing these details, users can proceed with the payment.

A screenshot of a computer

Description automatically generated

Figure 3. - Placed order page

Figure 3.9 Placed order page: The page displays placed orders, where detailed information about the orders is shown. Here, the page provides details such as the shop name, order status, and payment status for the orders that the user has placed. On the right side, there is a navigation bar leading to pages such as user details, order history, and vouchers.

A screenshot of a computer

Description automatically generated

Figure 3. - Personal information page

Figure 3.10 Personal information page: The user information page displays basic details about the user, including their username, name, phone number, address, gender, date of birth, and profile picture. On the left side, there is the user's email, which may need verification for added security.

A screenshot of a computer

Description automatically generated

Figure 3. 11 - User management page

Figure 3.11 User managementpage: User management page, where the admin can oversee all user accounts, including admin, sellers, and customers. The admin has the ability to modify corresponding permissions and access details such as email, full name, phone number, date of birth, gender, and role. Additionally, the admin can export the information to an Excel sheet if necessary.

A screenshot of a computer

Description automatically generated

Figure 3. 12 - Category management page

Figure 3.12 Category managementpage: Category management page, where the page displays existing categories with information such as category name and code. Above, there is a search bar used to search for categories. The admin can add, edit, and delete categories with existing information and can export the data to an Excel file if necessary.

A screenshot of a computer

Description automatically generated

Figure 3. 13 - Brand management page

Figure 3.13 Brand managementpage: Brand management page, where the admin can manage information about existing brands on the website, including details such as name and code. Above, there is a search bar used to search for brands. The admin can add, edit, and delete necessary information and can export the data to an Excel file if necessary.

A screenshot of a computer

Description automatically generated

Figure 3. 14 - Product management page

Figure 3.14 Product managementpage: Product management page, where the admin can manage information about existing products on the website, including details such as product name, category, brand, material, manufacturer, product views, and product status. Above, there is a search bar used to search for products. The admin can add, edit, and delete product information and can export the data to an Excel file if necessary.

A screenshot of a computer

Description automatically generated

Figure 3. 15 - Banner management page

Figure 3.15 Banner managementpage: Banner management page, where the admin can manage banners that will appear on the homepage. It includes details such as name and displayed image. Above, there is a search bar used to search for banners. The admin can add, edit, and delete necessary information and can export the data to an Excel file if needed.

A screenshot of a computer

Description automatically generated

Figure 3. 16 - Theme management page

Figure 3.16 Theme managementpage: Theme management page, where the admin can manage themes that will be displayed on the website. It includes details such as name and code. Above, there is a search bar used to search for themes. The admin can add, edit, and delete necessary information and can export the data to an Excel file if needed.

A screenshot of a computer

Description automatically generated

Figure 3. 17 - Shiping type management page

Figure 3.17 Shipping type managementpage: Shipping type management page, where the admin can manage different shipping methods. It includes details such as the name of the shipping method and the associated cost. Above, there is a search bar used to search for shipping methods. The admin can add, edit, and delete shipping methods and can export the data to an Excel file if needed.

A screenshot of a computer

Description automatically generated

Figure 3. - Supplier management page

Figure 3.18 Supplier managementpage: Supplier management page, where the admin can manage suppliers providing products for the website. It includes details such as the supplier's name, phone number, email, and address. Above, there is a search bar used to search for existing suppliers. The admin can add, edit, and delete necessary information and can export the data to an Excel file if needed.

A screenshot of a computer

Description automatically generated

Figure 3. - Inventory management page

Figure 3.19 Inventory managementpage: Inventory management page, where the admin can manage items to be imported into the warehouse. It includes detailed information such as the date of import, supplier name, phone number, and the name of the employee (with their position) responsible for importing the order. Above, there is a search bar used to search for import orders. The admin can view, add new import orders, and export the data to an Excel file if needed.

A screenshot of a computer

Description automatically generated

Figure 3. - Order management page

Figure 3.20 Order managementpage: Order management page, where the admin can oversee orders placed by users on the website. It includes information such as the user's phone number, email, order date, shipping type, voucher code, payment method, status, and assigned shipper. Above, there is a search bar used to search for orders. The admin can view detailed information about each order and can export the data to an Excel file if needed.

CONCLUSIONS AND SUGGESTIONS

##### Conclusions

Achievement:

* Understanding e-commerce operations with Blockchain integration:
* Gain an understanding of how an e-commerce website works when integrated with Blockchain.
* Integrate virtual currency payments into e-commerce applications.
* Successfully integrated virtual currency payments into e-commerce applications, introducing Blockchain to a wider audience.
* Successfully showcases the benefits of combining Blockchain with existing applications, highlighting key features.

Through this project, we have gained valuable insights into Blockchain and its integration into e-commerce. This is a relatively underexplored topic because Blockchain technology has not been widely adopted. In the age of technological advancement, integrating Blockchain into collaborative projects can benefit many people. Blockchain support can help e-commerce businesses demonstrate transparency about the quality of the products they offer. In turn, this streamlines product supply procedures, reduces transaction processing and product shipping costs, and even speeds up time-to-market for new products. Ultimately, it creates opportunities to expand the market and build trust in reaching consumers online.

##### Suggestions

* Promote Blockchain technology to a wider audience, especially in the evolving technology landscape. Highlight the benefits of applying Blockchain to ongoing projects.
* Conduct further research to explore new features and enhance existing ones in Blockchain technology.
* User-friendly interface design simplifies the complexity of Blockchain for both businesses and e-commerce users. This ensures a smooth transition to blockchain technology without requiring extensive technical knowledge.
* Establishing industry standards for blockchain interoperability. This enables seamless communication and data sharing between different e-commerce platforms and blockchain networks.
* Encourage collaborative cooperation between e-commerce platforms and blockchain developers. Joint initiatives can lead to innovative solutions and faster adoption across the industry.

REFERENCES

1. Stackoverflow.com
2. Google.com
3. Github.com
4. W3schools.com
5. React.dev
6. Github.com