

VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY  
INTERNATIONAL UNIVERSITY  
INFORMATION TECHNOLOGY - DEPARTMENT OF COMPUTER SCIENCE



---

## WEB APPLICATION DEVELOPMENT

---

### Project report E-COMMERCE WEB-BASED APPLICATION INTEGRATED WITH MOMO PAYMENT METHODS

---

Lecturer: Assoc.Prof. Nguyen Van Sinh  
Team: HuongHoa

No	Student name	Student ID
1	Nguyen Thi Mai Huong	ITITIU19128
2	Ho Thi Thu Hoa	ITITIU19120



## TABLE OF CONTENTS

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Overview . . . . .	5
1.2	Project objectives and scope . . . . .	5
<b>2</b>	<b>System analysis and design</b>	<b>6</b>
2.1	Limitations of traditional shopping . . . . .	6
2.2	Our proposed system . . . . .	6
2.2.1	System functional requirements . . . . .	6
2.2.2	System non-functional requirements . . . . .	10
2.2.3	Resources requirements . . . . .	11
2.3	System artifacts . . . . .	12
2.3.1	Use case diagram . . . . .	12
2.3.2	Database analyze and design . . . . .	13
<b>3</b>	<b>Applied tools, techniques, and frameworks</b>	<b>15</b>
3.1	ReactJS . . . . .	15
3.1.1	Core Features of React . . . . .	16
3.2	NodeJS . . . . .	17
3.2.1	Introduction about NodeJS and its operation method . . . . .	17
3.2.2	NodeJS advantages . . . . .	17
3.3	MySQL . . . . .	18
<b>4</b>	<b>Demonstration</b>	<b>19</b>
4.1	Register/Login/Forgot password Layout . . . . .	19
4.2	Client UI . . . . .	22
4.3	Admin UI . . . . .	26
<b>5</b>	<b>Conclusion</b>	<b>29</b>
5.1	Short Retrospective . . . . .	29



5.1.1	Lesson learned . . . . .	29
5.1.2	Limitations in the project . . . . .	30
5.2	Conclusion . . . . .	30
<b>6</b>	<b>References</b>	<b>31</b>



## LIST OF FIGURES

1	The overall modules included in our e-commerce web-based system . . . . .	12
2	The integration of summary goal use case and user goal use case of the system .	13
3	The Entity-Relationship Model of sytem . . . . .	14
4	Extracted object-relational mapping diagram . . . . .	15
5	Login layout . . . . .	19
6	Forgot password layout . . . . .	20
7	Register layout . . . . .	21
8	Header layout . . . . .	22
9	Footer layout . . . . .	22
10	Load product list layout . . . . .	22
11	Showing the product list sort by category . . . . .	23
12	Product detail layout with showing a list of related products . . . . .	24
13	The shopping cart layout . . . . .	24
14	Fill receiver and payment method information when a user process payment . .	25
15	The payment process of our system . . . . .	25
16	The payment status layout displayed after a user process an payment of our system	26
17	The analytic layout in the admin side . . . . .	27
18	The Users management layout in the admin side . . . . .	27
19	The Products management layout in the admin side . . . . .	28
20	The Products management layout when loading many products in the admin side	28
21	The Orders management layout in the admin side . . . . .	29



## 1 Introduction

The era of digital technology 4.0 along with the strong development of the internet has promoted the development of many economic sectors in the world in general as well as in Vietnam particularly. In particular, the e-commerce industry has become a trend and gradually replaces the traditional trading market when it not only helps buyers easily select products, compare their prices in the fastest way, and buy almost anything remotely just by simple operations, but also helps sellers bring their items to many domestic and foreign customers. Based on that situation, our team has built an e-commerce platform with a variety of items such as clothes, shoes, and jewelry on the website platform in the form of payment in cash or electronic wallet momo.

### 1.1 Overview

Nowadays, many fashion brands have been very successful in maintaining and expanding their online business thanks to the website being efficient and beneficial for businesses and shoppers. ***From the customer's perspective***, they can easily compare the price, check the product's quality with reliable feedback, receive products at home, or efficiently hunt attractive "sale off" programs. ***For companies***, e-commercial bring remarkably a variety of benefits such as access to more potential customers, saving staff costs, advertising, and easy payment with the Customer's bank account. Based on the significant advantages of Online Shopping Platforms, we decided to make a web-based application in the fashion business. Since our main customers using this application are young people and men, user-friendly interaction and focusing on customer experience are our major priorities.

Our project is an ***e-commerce web-based application integrated with momo payment*** that allows clients to find our products on our website and quickly buy them and easily make online payments via momo wallet. This makes it easy for our customers to control their spending. Our goal is to build the most convenient website for our customers.

### 1.2 Project objectives and scope

The main objective of our fashion website project is to manage products in the shop, customer accounts, and orders the client purchased. It manages all the information related to Customers, Products, Orders, and Categories that the shop has. The purpose of the project is to build an application program to reduce the manual work for managing the e-commerce website, Product, Customer, Product Type, etc.

Our system scope aim for a local fashion shop. We are creating an online portal where their customers can enjoy easy shopping from anywhere. Meanwhile, customers just need to surf our website in order to view available products, select items, complete orders with momo payment,



and receive their fashion items through home delivery companies.

## 2 System analysis and design

### 2.1 Limitations of traditional shopping

Shopping directly at fashion stores to choose the most suitable with clients in a traditional way exists a variety of disadvantages, such as:

- ***Geographical limitations and inconvenience:*** Traditional shopping requires customers to visit physical stores that are remarkably limited by geography. This can be inconvenient for those in remote areas or without transportation, leading to customers' time-consuming processes, or might not be in reach of distant clients.
- ***Impersonal customer service and limited assistance:*** Traditional shopping may offer less personalized customer service due to limited staff availability. Clients might get some advice just from the seller, which might not make them satisfied when compare to refer a wide range of feedback on shopping online.
- ***Inflexible shopping hours and peak period challenges:*** Physical fashion stores have fixed operating hours, limiting shopping opportunities for individuals with busy schedules. Besides, shopping at big shops in peak periods might be frustrating because of crowds, long queues, and parking difficulties.
- ***Less user-friendly in some cases and limited access to information:*** Traditional shopping lacks the convenience of easily accessible detailed product information, customer reviews, and recommendations available online. Therefore, customers cannot refer to similar items' prices from other places to compare whether it's reasonable or not.

### 2.2 Our proposed system

In order to address several existing problems of traditional shopping, our e-commerce web-based system allows clients to select and order products online through various items. The system will manage the customers and product orders. It also recommends a home delivery system for the purchased products, and clients can make payments online for orders via Momo wallet.

#### 2.2.1 System functional requirements

In our project, due to time and using frameworks ability limitation we just handle some basic ***functional requirements*** such as CRUD for each main entities handled in the system:



Name	Register/Login/Forgot password
Detail	<ul style="list-style-type: none"><li>– A new user will have to register in the system by providing essential details in order to add the products in the system into his cart.</li><li>– Registration of clients plays an indispensable role when clients purchase items. It helps the Admin get the customer's information so that they can complete the customer's payment and delivery.</li><li>– Forgot Password function helps users create new passwords.</li></ul>
Rationale	Registration helps the shop take customer information in order to not only serve for shipping customers' orders but also provide the information of searching as well as buying products to discover the best suitable items for customers in the next purchase and find prevalent products based on the orders at the website.
Requirements	<p>After the user registers, logs in, or resets the password, all user information will be saved in a database</p> <ul style="list-style-type: none"><li>• Username and password will be provided after user registration is confirmed.</li><li>• System must be able to verify and validate the information.</li><li>• Password should be hidden in default from others while typing it in the field or can be shown if the user clicks the eye icon.</li></ul>
References	Login, register, validate, and forgot password page.

Name	View Products
Detail	A variety of products exist in the shop's database with information such as name, price, images, etc will be loaded on the website UI
Rationale	This function helps shop show all available products visible and helps clients view all of them.
Requirements	Customer can see any items on the web page
References	Home page, Products Details page, Admin page



Name	View Product's Details
Detail	<ul style="list-style-type: none"><li>– When the customer clicks on any item, he will see details of that product and a list of related items.</li><li>– He can add this item to his cart by clicking the button “Add to cart”</li></ul>
Rationale	<p>Customer can choose any product to view more detailed information</p> <ul style="list-style-type: none"><li>– A product has to fulfill information such as images, accurate price, description, rating, etc there exists a function for the customer to choose an item, and increase or decrease the number of items that the client need.</li><li>– He can add this item to his cart by clicking the button “Add to cart”</li></ul>
Requirements	This page must be displayed exactly the information of the item, which clicked by the customer.
References	Home page, Products Details page, Admin page



Name	Add to cart / Purchasing items
Detail	<ul style="list-style-type: none"><li>– The user can add the desired product to his cart by clicking add to cart option on the product. He also can view his cart by clicking on the cart button. All products added by the cart can be viewed in the cart.</li><li>– Users can remove an item from the cart by clicking the icon remove.</li><li>– After confirming the items in the cart the user can submit the cart by providing a delivery address. On successful submission, the cart will become empty.</li></ul>
Rationale	<ul style="list-style-type: none"><li>– It helps customers select all of their desired products and take time to thoroughly consider without worrying forgot any item they prefer before by saving their data cart in the system web page.</li><li>– Customers can own easily their preferred items by payment online without wasting time.</li></ul>
Requirements	<ul style="list-style-type: none"><li>– The system must ensure that only a registered customer can purchase items.</li><li>– Payment system must be exactly accurate in all of the information related.</li></ul>
References	Shopping Cart and Payment page

Name	Search engines
Detail	The search engine for items does not require all the words.
Rationale	When a client surfs our web page, he sometimes wants to search exactly one or a few items they need, and maybe they can type lacking in the filter box, search engines will help them to find out what they want better and faster.
Requirements	When a client enters the text into the filter field, the system will perform the statement with the LIKE operator on the database, and filter the products containing characters the client typed.
References	Home and Admin page



Name	Add/Update products/ Add category/ View customers information
Detail	Administrator can: <ul style="list-style-type: none"><li>• Add a category.</li><li>• Add/update a product.</li><li>• View customers information</li><li>• Manage total income/ customers/ products/ orders.</li></ul>
Rationale	It helps administrators easily manage and interact with the shop's database via front-end GUI.
Requirements	The system must identify the login of the admin Admin account should be secured so that only the owner of the shop can access that account.
References	Admin page

### 2.2.2 System non-functional requirements

Alongside handling functional requirements, we also take into account addressing several ***non-functional requirements*** that can enhance the overall behavior and qualities of the system:



Requirement	Description
Efficiency	Client can effectively access the system as finding and purchasing products would be easy and convenient
Dependable	The system ought to precisely perform payment, product cost, and quality, etc
Performance loading	The system is optimized to load quickly and respond promptly to user interactions by ensuring efficient rendering and minimizing network requests on the FE side, in the meanwhile, the server side also be optimized by efficiently handling database queries, etc
Security	Our website has a basic authentication algorithm to secure and protect user data from unauthorized access, manipulation, or breaches.
Reliable	The system provides a reliable environment to both customers and administrators. All orders should be reaching the admin without any errors.
Usable	The system is intended for an easy to the most suitable product so that we can make users' experience more enjoyable, satisfying, and personalized on the basis of their needs.
Maintainability	The website's codebase is easy to understand, modify, and maintain over time. We use the most well-known framework currently, which are ReactJs and ExpressJS for front-end and back-end respectively.

### 2.2.3 Resources requirements

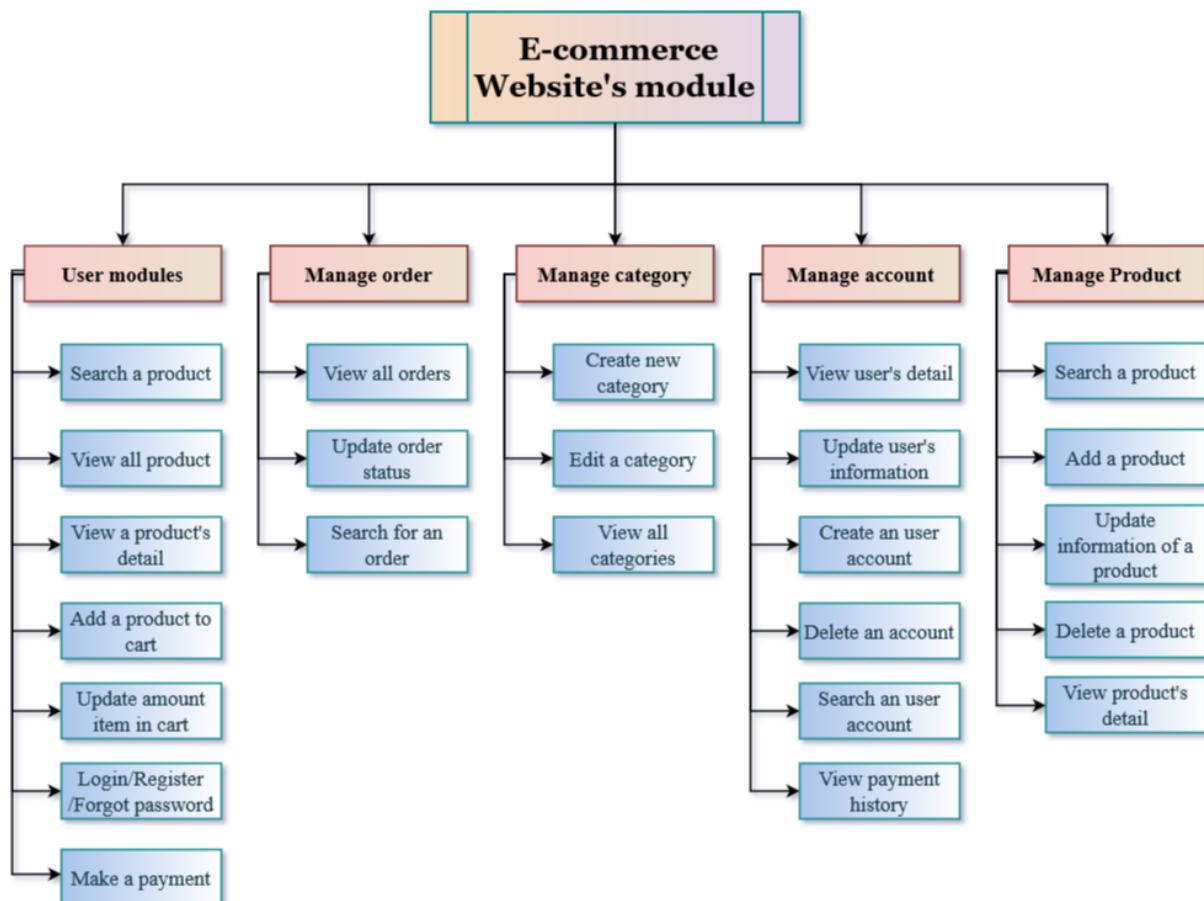
In terms of resource types that are required in our system, it includes 2 main categories **hardware resources** and **software resources**.

- **Hardware resources:** Since all types of computers with the necessary software installed can be used to implement the program, I will not mention much about it. Each member of our group is required to use the personal computer/ laptop in all the processes from studying to implementing and testing software components.
- **Software resources:** Implementing components of our project requires each team member to install several software, such as:
  - Visual studio code: coding the system
  - Xampp using phpMyAdmin(mySQL): manage the database
  - MSEExcel: collect data from real pages online
  - MS Windows 10 Home 64bit: Operate system

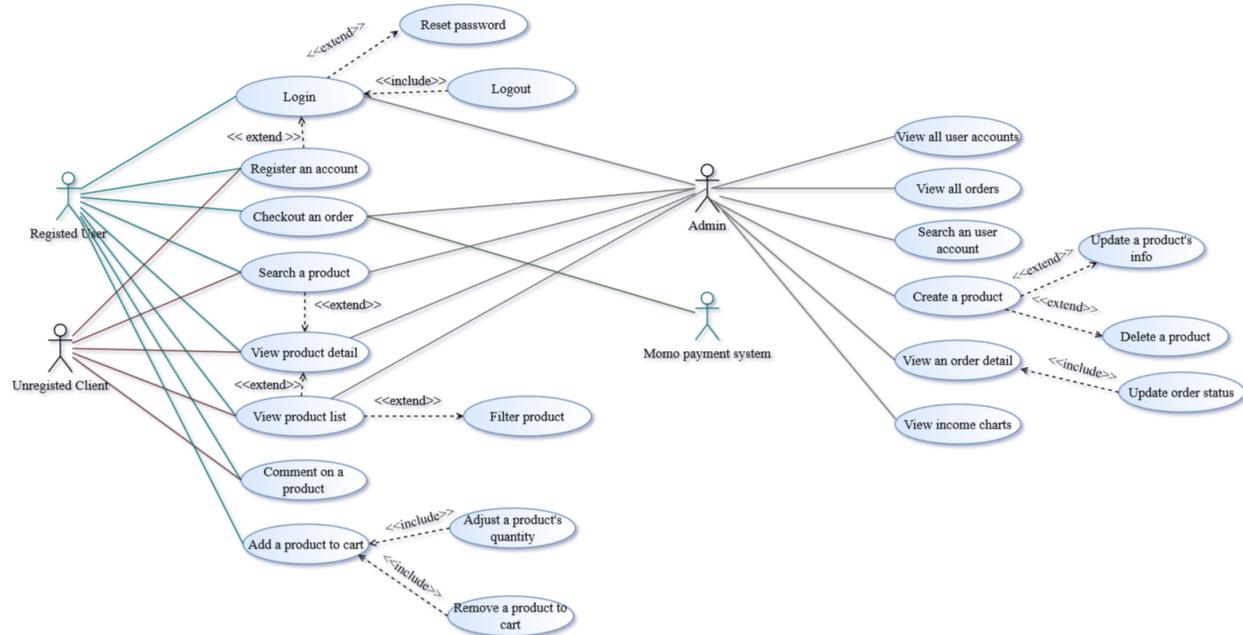
## 2.3 System artifacts

### 2.3.1 Use case diagram

Based on the functional requirement that we already analyzed in Section 2.2.1, we design our system functions followed by these below system modules and use case diagrams respectively:



Hình 1: The overall modules included in our e-commerce web-based system



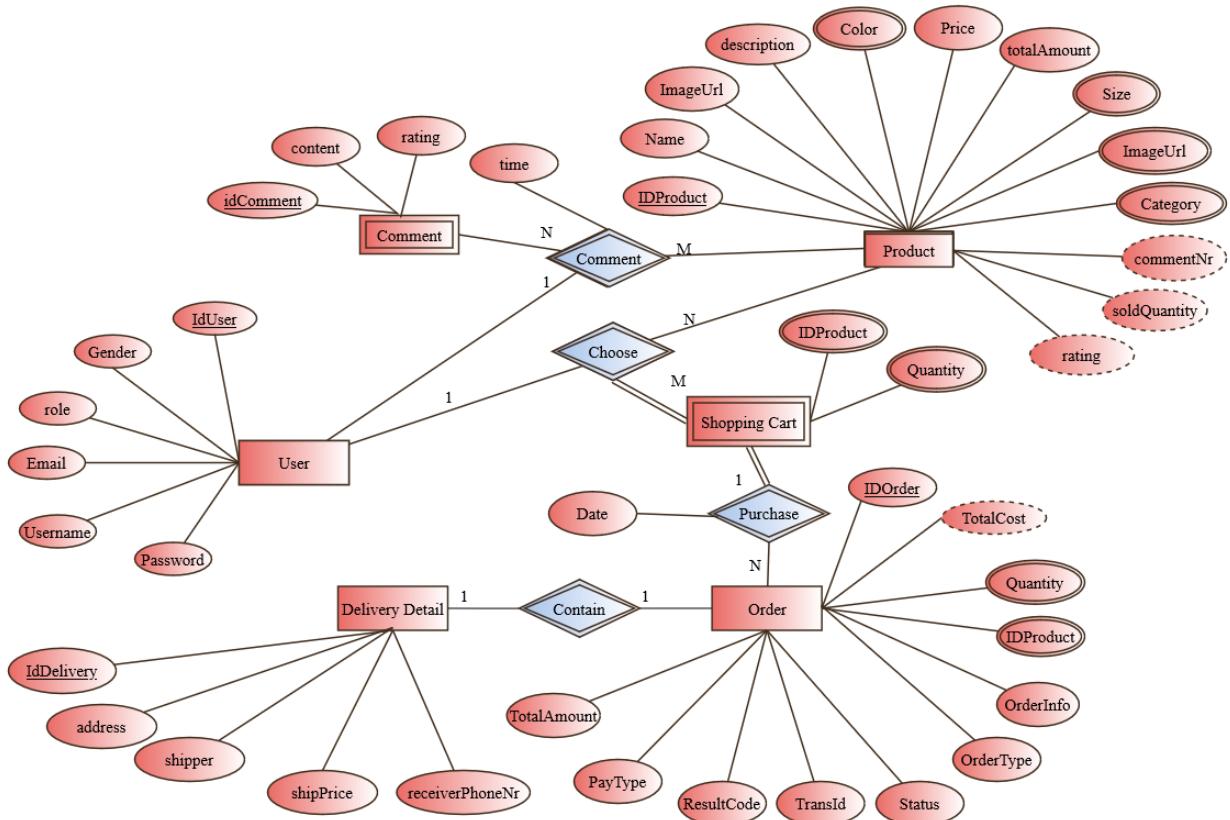
Hình 2: The integration of summary goal use case and user goal use case of the system

### 2.3.2 Database analyze and design

With the analysis details above, we decided to design the database with 6 main entities in our system, which are Product, User, Shopping cart, Order, Comment, and Category. They do have relations with each other.

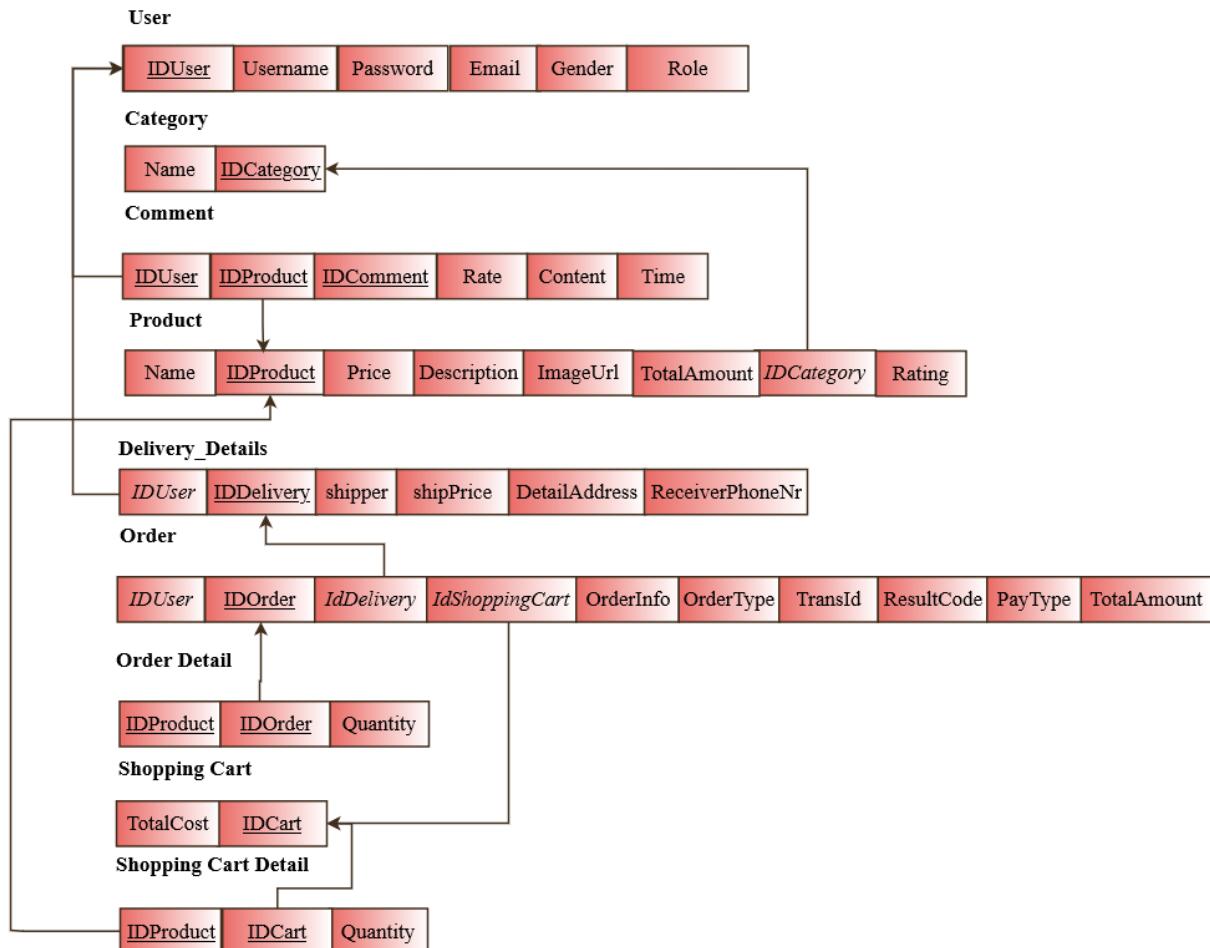
- A **User** can make many **comments** on a **Product**, and a product also can have multiple comments from several Clients. However, each comment must belong to a user and specific product. Therefore, the relationship between User - Product - and Comment is a 1:M: N ternary relationship.
- A Registered **User** also can select some **products** from the website UI to his/her **shopping cart**. Each product can be in many carts at the same time. However, each shopping cart must belong to a user. Therefore, the relationship between Shopping Cart - User - and Product is a 1:M: N ternary relationship.
- Each **order** of a user will include products that must be selected in the **Shopping Cart**. 1 Shopping Cart can have multiple Orders from a user. So, the relationship of Shopping Cart and Order is 1-N relation
- Each order will be sent to the User to a receive address that the client provides when he/she makes an order. So an **Order** will have a **delivery detail** with a 1:1 relation.

==> Therefore, we have an Entity Relationship diagram as shown below:



Hinh 3: The Entity-Relationship Model of system

Apply normal form rules on ERD, we get the object-relational mapping diagram (ORM):



Hình 4: Extracted object-relational mapping diagram

### 3 Applied tools, techniques, and frameworks

In our project we used ReactJS for front-end implementation and ExpressJS of NodeJS to implement the backend side. We use MySQL for the database part.

#### 3.1 ReactJS

ReactJS is not a JavaScript framework. That's because it's only responsible for rendering the components of an application's view layer. React is an alternative to frameworks like Angular and Vue, which all allow the creation of complex functions.

React work method:



- One of the biggest advantages of using React is that you can infuse HTML code with JavaScript.
- Users can create a representation of a DOM node by declaring the Element function in React.
- A React app usually has a single root DOM node. Rendering an element into the DOM will change the user interface of the page.
- Whenever a React component returns an element, the Virtual DOM will update the real DOM to match.

### 3.1.1 Core Features of React

React has some core features that make it stand out from other JavaScript libraries:

- Earlier creating a dynamic web page required a lot of complex coding. It needed specific HTML strings for dynamic web pages. Using ReactJS is no more an issue. It makes use of JSX which has a particular syntax and lets HTML quotes and tags render particular sub-components. In addition to this, it also supports building machine-readable code and mixes components into an individual variable file. In React regular JavaScript is not used for creating its templates.
- **Virtual DOM:** The Document Object Model (DOM) presents a web page in a data tree structure. ReactJS stores Virtual DOM trees in the memory. By doing so, React can apply updates to specific parts of the data tree, which is faster than re-rendering the entirety of the DOM tree. Whenever there's a change in data, ReactJS will generate a new Virtual DOM tree and compare it with the previous one to find the quickest possible way to implement changes in the real DOM. This process is known as diffing. By making sure that UI manipulation only affects specific sections of the real DOM tree, rendering the updated version takes less time and uses fewer resources. The practice greatly benefits large projects with intense user interaction.
- **Components and Props:** ReactJS divides the UI into isolated reusable pieces of code known as components. React components work similarly to JavaScript functions as they accept arbitrary inputs called properties or props.
- **State Management:** A state is a JavaScript object that represents a part of a component. It changes whenever a user interacts with the application, rendering a new UI to reflect the modifications. State management refers to the practice of managing React application states. It includes storing data in third-party state management libraries and triggering the re-rendering process each time data is changed.



- Programmatic Navigation: Programmatic navigation refers to instances when lines of code create an action that redirects a user. Login and sign-up actions, for instance, programmatically navigate users to new pages. React Router, React's standard library for routing, provides multiple ways of safe programmatic navigation between components without requiring the user to click on a link. Using a Redirect component is the primary method of programmatic navigation with history.push() is another approach.

*The React Router package synchronizes the UI with the URL, giving control over the look of React applications without depending on links.*

## 3.2 NodeJS

### 3.2.1 Introduction about NodeJS and its operation method

Nodejs is an independent development platform (Platform) built on V8 JavaScript Engine – an interpreter that executes JavaScript code that makes it possible to build web applications such as video clips, forums and especially is a narrow social networking site that quickly and easily expands. Node.js has been built and developed since 2009, sponsored by Joyent company.

NodeJS can run on many different operating system platforms from Windows to Linux, and OS X so that is also an advantage. NodeJS provides rich libraries in the form of various Javascript Modules that simplify programming and reduce time to a minimum.

*NodeJS's operation method:*

- The main idea of Node js is to use non-blocking, directing data input and output through real-time tasks quickly. Because Node js is rapidly scalable and capable of handling a large number of concurrent connections by high throughput.
- If in traditional web applications, requests create a new request processing thread and occupy the system's RAM, the system's resources will be used inefficiently. Therefore, the solution that Node js offers is to use single-threaded (Single-Threaded), combined with non-blocking I/O to execute requests, allowing tens of thousands of concurrent connections to be supported.

### 3.2.2 NodeJS advantages

NodeJS is widely applied in many enterprise projects because it meets many different uses:

- NodeJS does not need to wait for the API to return data, so any APIs in the NodeJS library are not synchronized.



- NodeJS is a Platform, not a Framework. Therefore, Node JS allows you to build websites independently and faster.
- NodeJS can run on multiple platforms including Windows, MacOS, and Linux.
- NodeJS is considered a single-threaded server and cannot support multithreading.
- NodeJS is not considered a programming language, so newbies must have a solid grasp of basic programming knowledge such as protocols, Javascript, etc. to be able to use NodeJS. However, the NodeJS community is usually very large, and ready to support you anytime, anywhere.
- NodeJS can create, open, read, write, delete, and close files while they are on the server.
- The core part of NodeJS is usually known in C++ language, so its performance and processing speed are relatively high. As a result, most NodeJS applications are capable of responding to real-time running on cross-platform, multi-device, etc.
- Build content for dynamic websites.
- Perform data collection according to specific requirements.
- Perform query, edit, delete, add data in basic management systems such as: Microsoft SQL Server, MySQL, MongoDB, PostgreSQL.

**In this project, I build APIs using the NodeJS framework**

### 3.3 MySQL

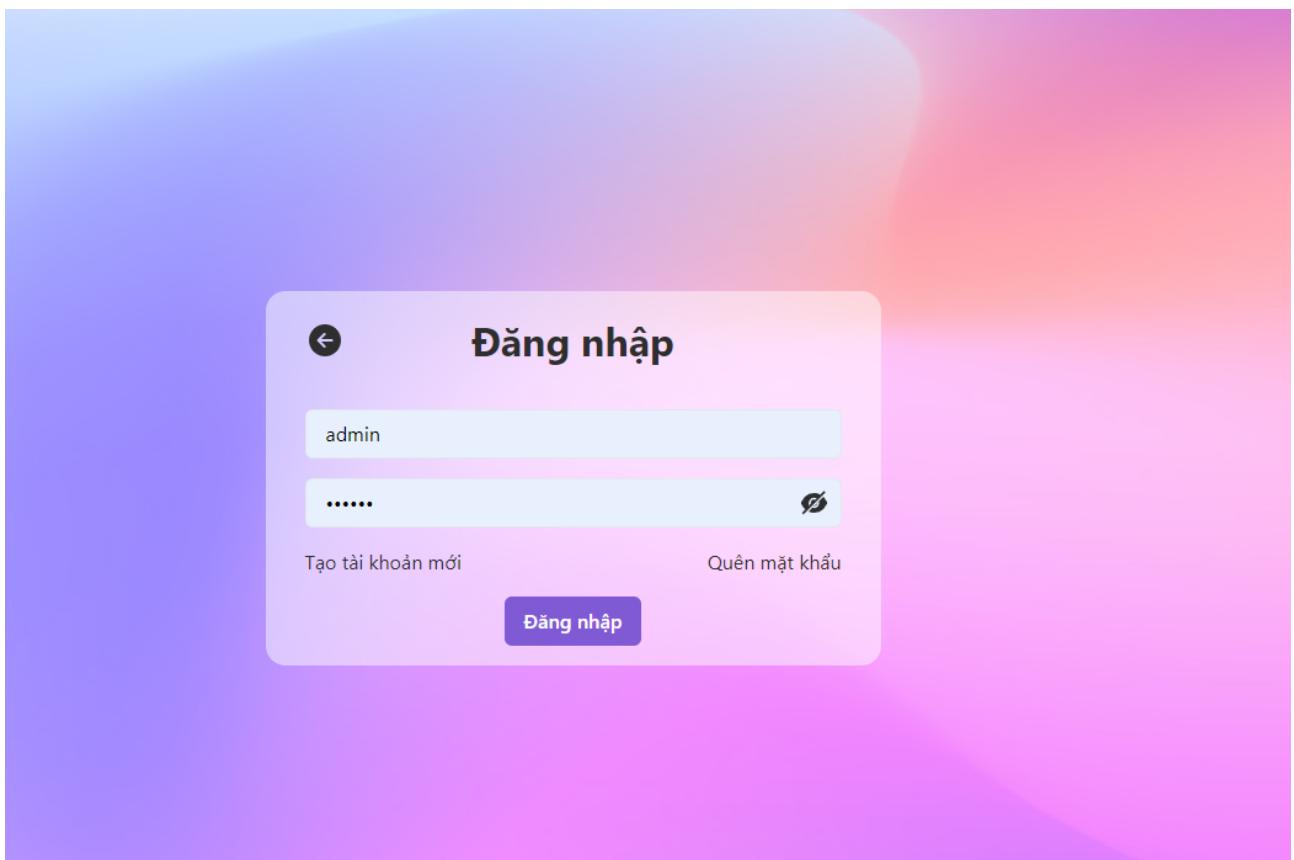
MySQL is the world's second most widely used open-source relational database management system (RDBMS). The SQL phrase stands for Structured Query Language. MySQL is the database management of the system or a database server. phpMyAdmin is a web application written primarily in PHP. It's used for managing the MySQL database. PhpMyAdmin intended to handle the administration of MySQL over the Web, and support most MySQL features:

- Browse and drop databases, tables, views, fields, and indexes.
- Create, copy, drop, rename, and alter databases, tables, fields, and indexes.
- Maintenance server, databases, and tables, with proposals on server configuration.
- Execute, edit, and bookmark any SQL statement, even batch queries.
- Manage MySQL user accounts and privileges.
- Manage stored procedures and triggers.



## 4 Demonstration

### 4.1 Register/Login/Forgot password Layout



Hình 5: Login layout



The diagram illustrates a two-step forgot password process. It consists of two rectangular panels side-by-side, each containing a form. A green border surrounds the left panel, and a red arrow points from the right side of the first panel to the second panel.

**Step 1: Xác thực thông tin** (Numbered 1)

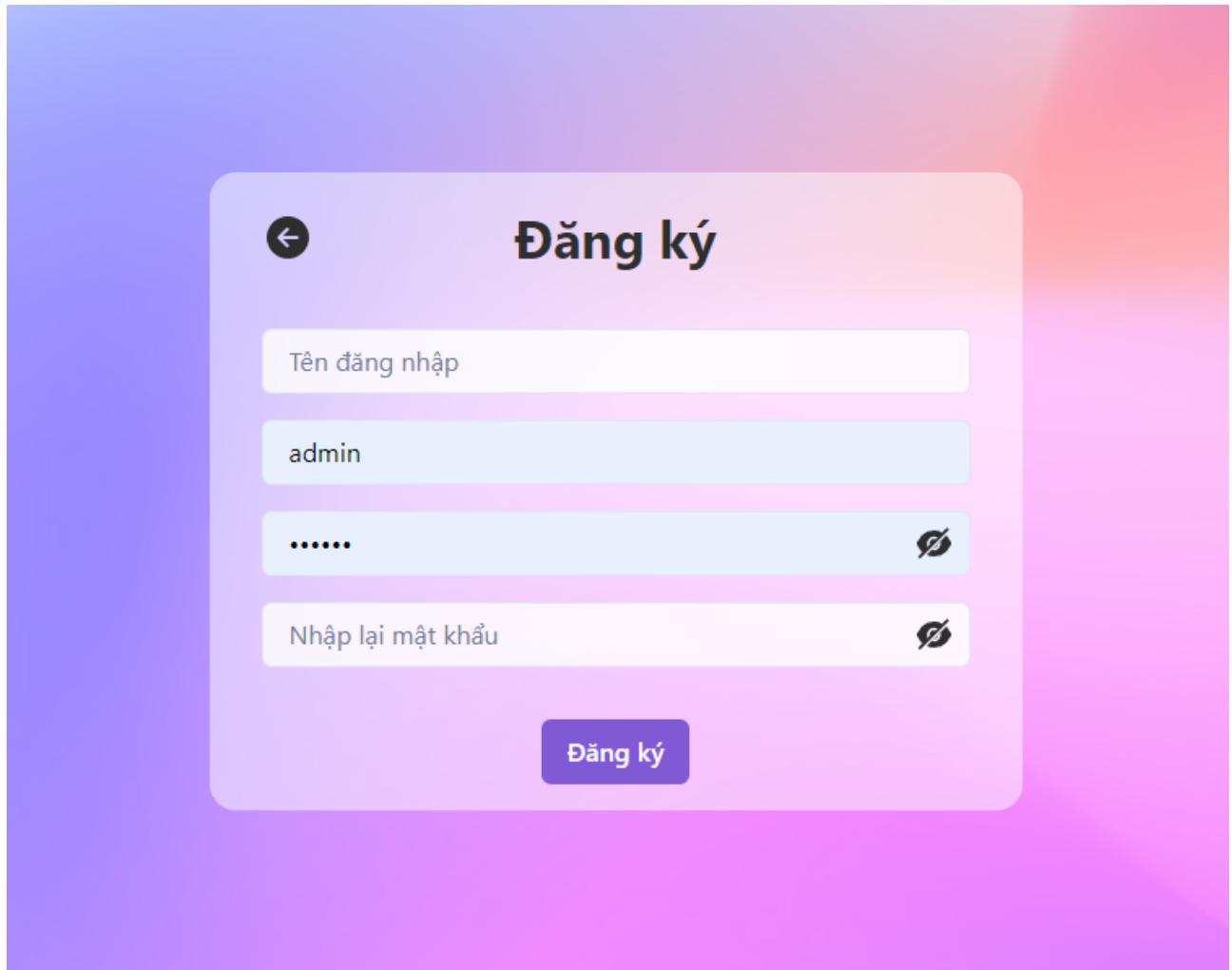
This panel contains fields for 'Tên đăng nhập' (Login name) and 'Email đăng ký' (Registered email), both with clear icons. Below these fields is a blue button labeled 'Xác thực' (Verify).

**Step 2: Đổi mật khẩu** (Numbered 2)

This panel contains fields for 'Mật khẩu mới' (New password) and 'Nhập lại mật khẩu' (Re-enter password), each accompanied by a clear icon. Below these fields is a blue button labeled 'Cập nhật' (Update).

A green callout box at the top right of the second panel indicates a successful action: 'Forgot' with the message 'Xác thực thành công!' (Verification successful!).

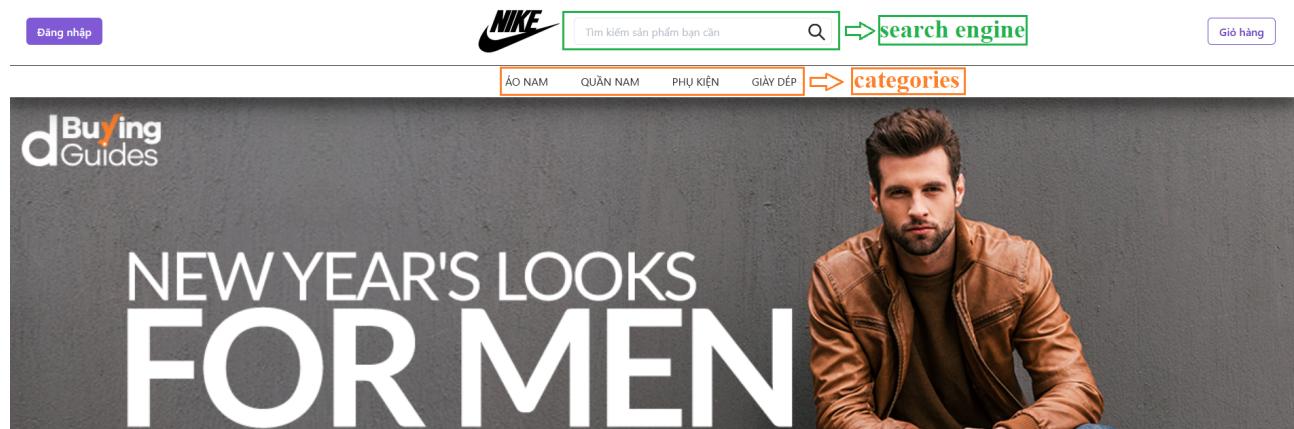
Hình 6: Forgot password layout



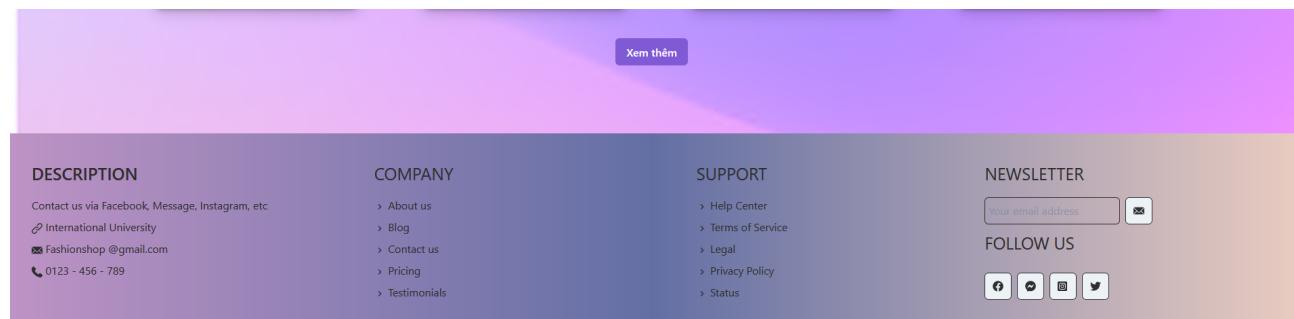
Hình 7: Register layout



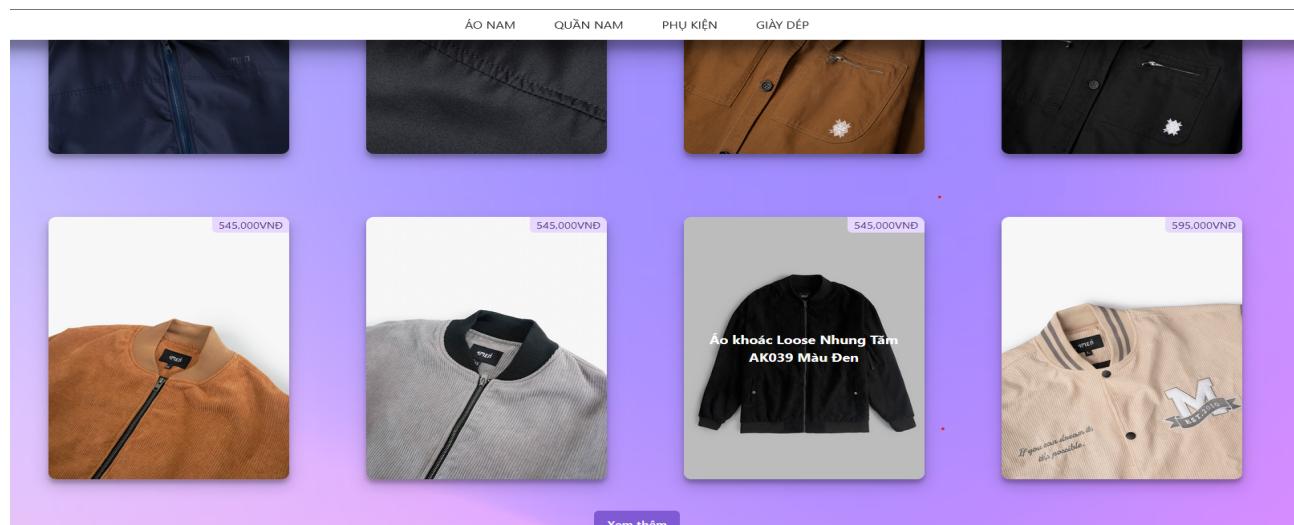
## 4.2 Client UI



Hình 8: Header layout



Hình 9: Footer layout

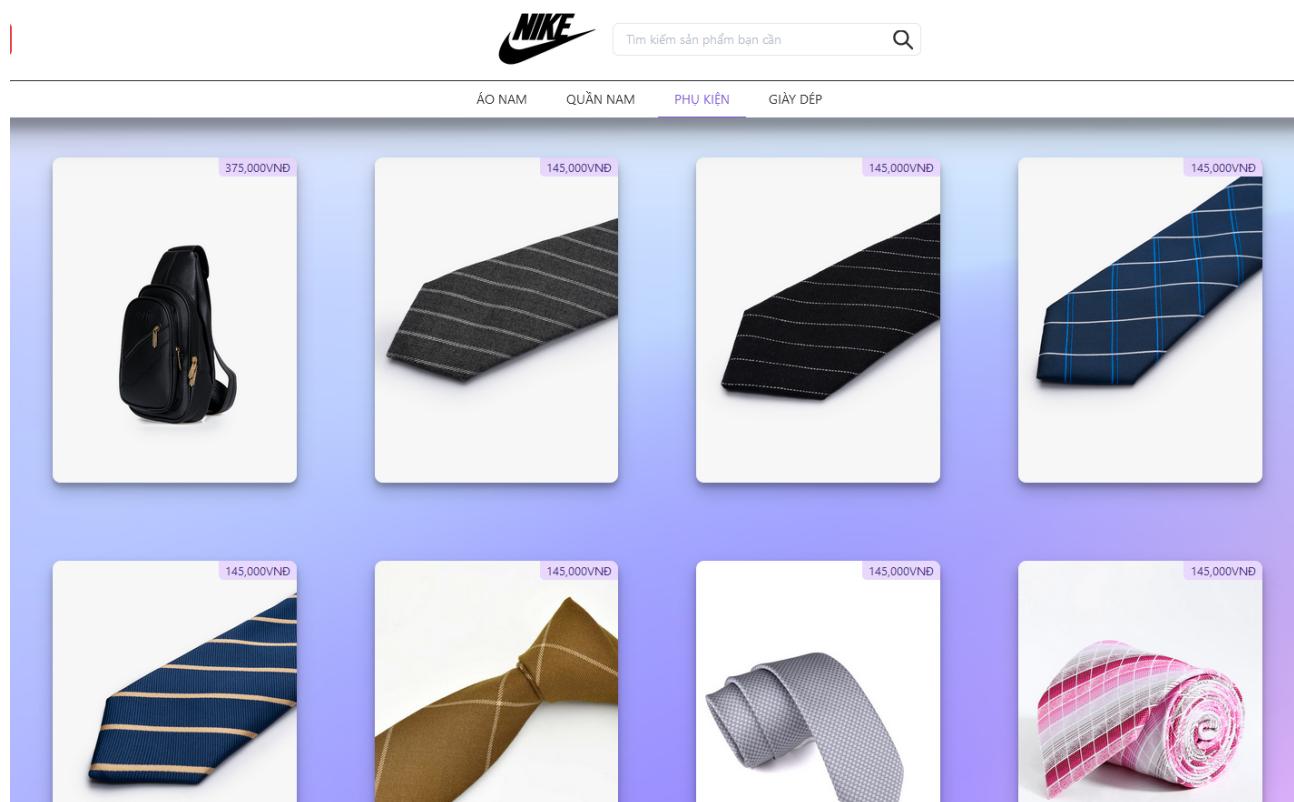


Hình 10: Load product list layout



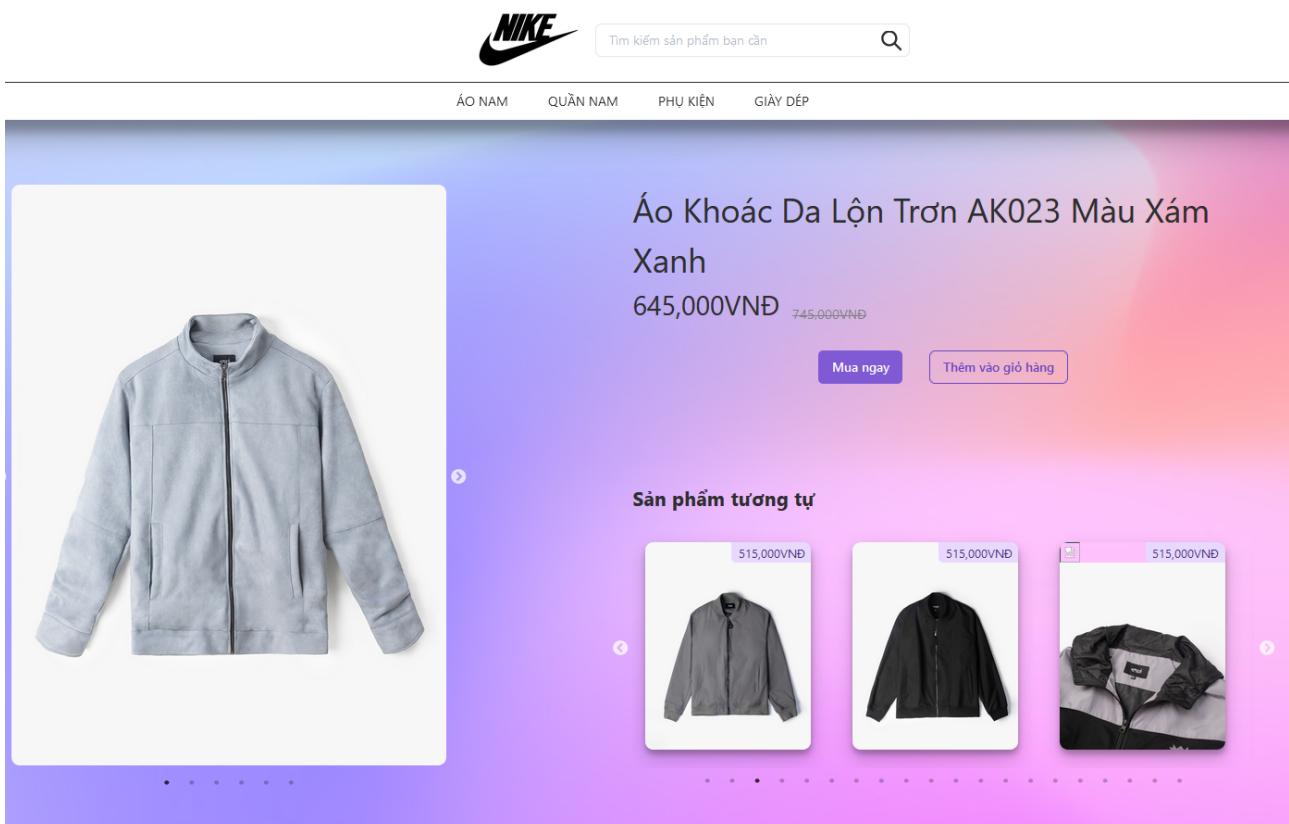
For the loading button of product list, whenever user press the "Xem them" button, system will load more products with limit number of each time load right below the old list. Just use the **Concat function** in the code.

Our system also allows users to view product lists by available categories, as the below figure section:

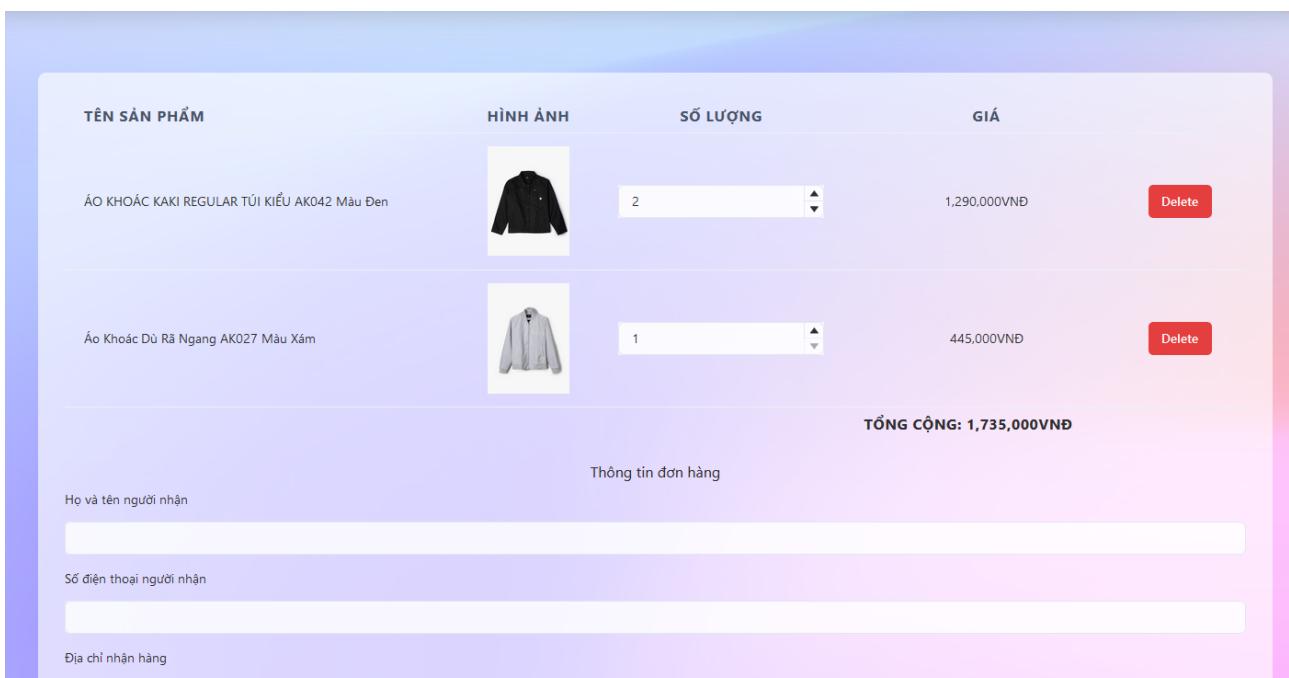


Hinh 11: Showing the product list sort by category

When a client clicks on a product, the system will automatically direct to the product detail of its products that show all its information and a list of related products with the same category.



Hình 12: Product detail layout with showing a list of related products



Hình 13: The shopping cart layout



Thông tin đơn hàng

Họ và tên người nhận  
Đào Linh Đan

Số điện thoại người nhận  
0369872838

Địa chỉ nhận hàng  
123, An Bình, Linh Xuân, Dĩ An

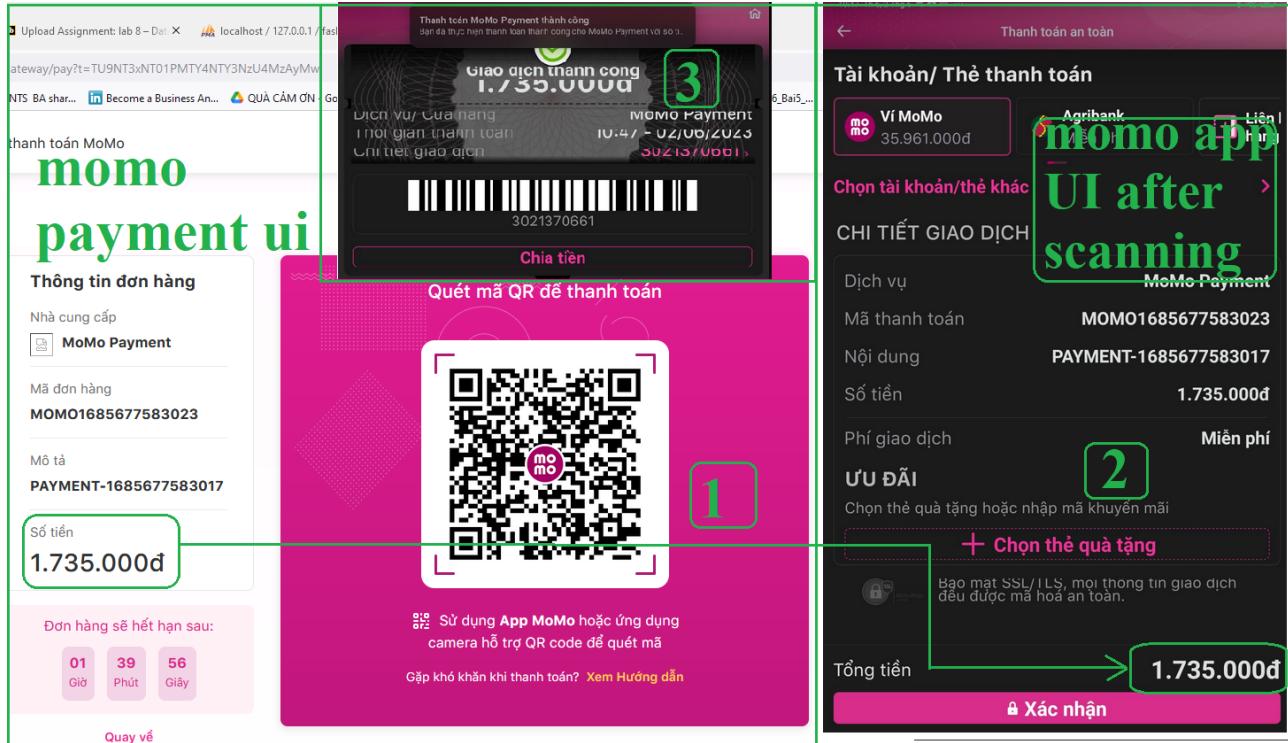
Đơn vị vận chuyển  
Giao hàng tiết kiệm

Ghi chú  
Đóng hàng kỹ giùm em nhà shop iuuu

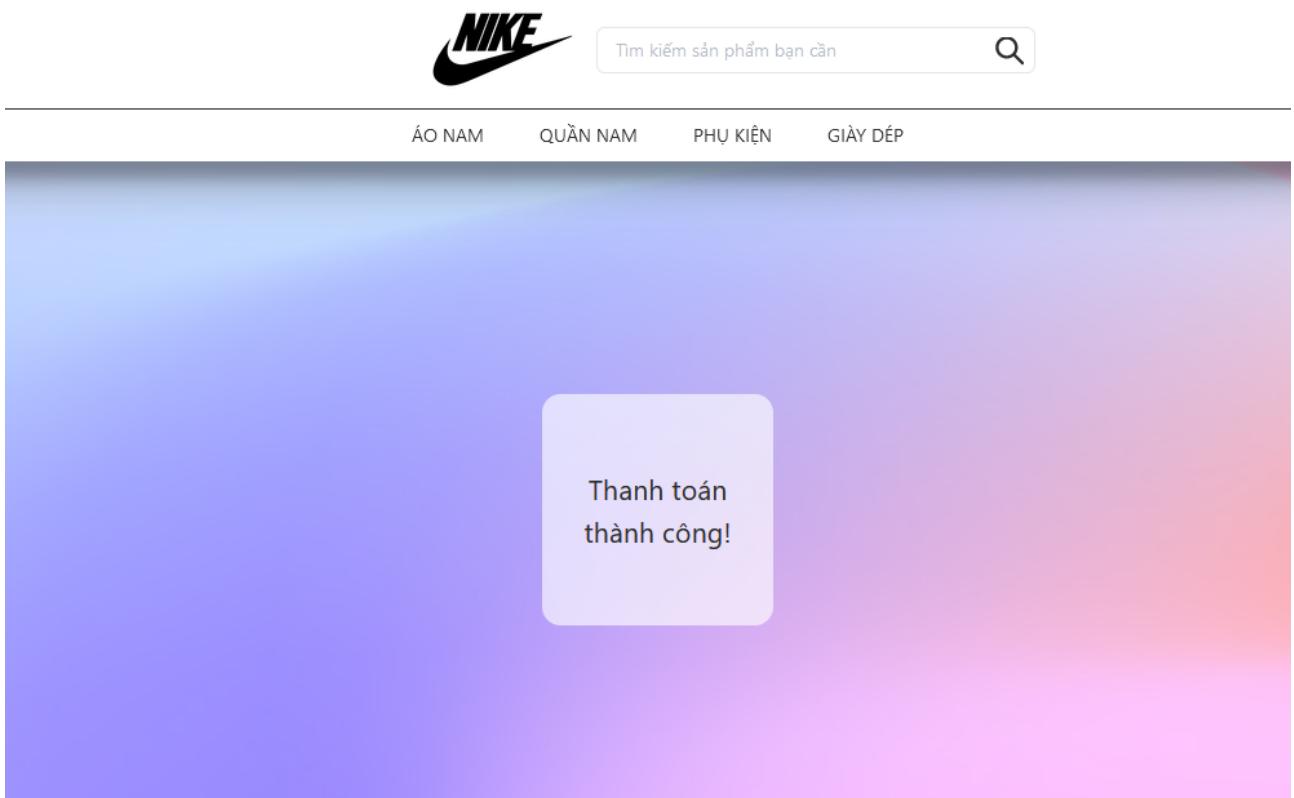
Phương thức thanh toán  
 Ví MoMo       Thẻ ngân hàng nội địa       Thẻ visa

[Tiến hành thanh toán](#)

Hình 14: Fill receiver and payment method information when a user process payment



Hình 15: The payment process of our system



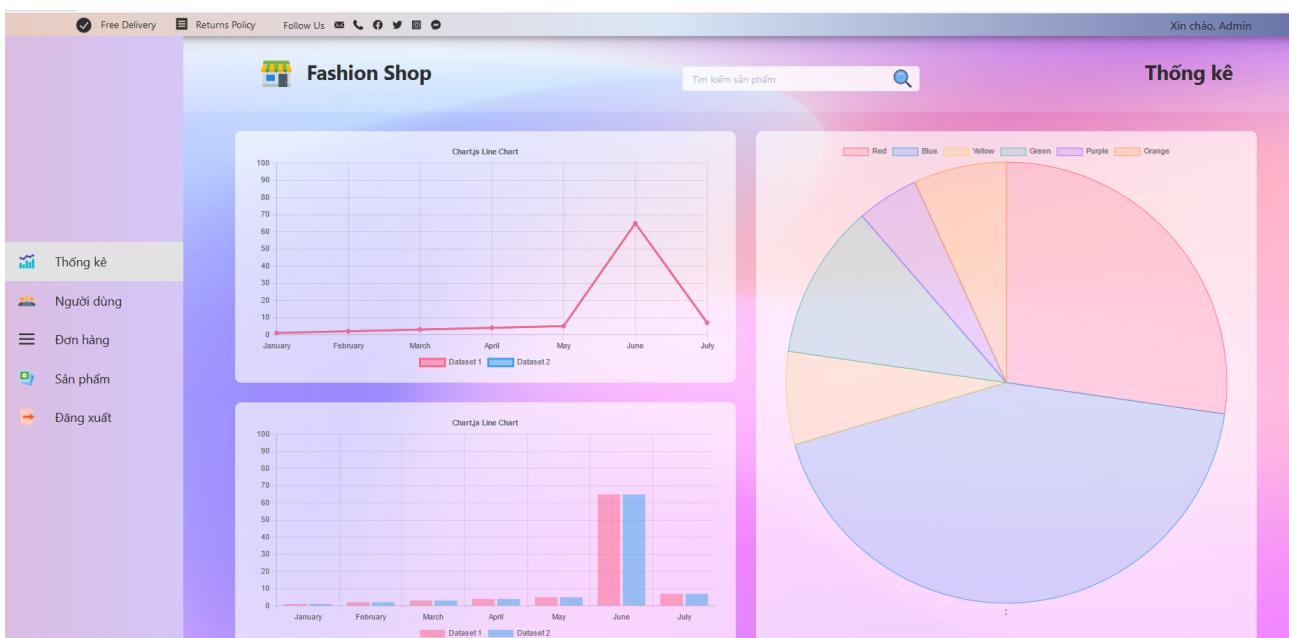
Hình 16: The payment status layout displayed after a user process an payment of our system

### 4.3 Admin UI

The analytic part contains three charts to represent the number of orders, the payment status percentages, and the income of the shop.



## International University Information Technology - Department Of Computer Science



Hình 17: The analytic layout in the admin side

The figure shows a dashboard titled "Fashion Shop" with a sidebar menu on the left. The menu items are: Thống kê (selected), Người dùng (selected), Đơn hàng, Sản phẩm, and Đăng xuất. The main area displays a table titled "Người dùng" listing 20 users:

MÃ NGƯỜI DÙNG	TÊN ĐĂNG NHẬP	EMAIL	PHẦN QUYỀN
1	admin	abc@gmail.com	Admin
2	user	sample@gmail.com	Người dùng
16	doantanloc3	abc@gmail.com	Người dùng
17	doantanloc	abc@gmail.com	Người dùng
18	abcd	doantanloc	Người dùng
19	abcdwe	abcd	Người dùng
20	nguyenthimy	abcd	Người dùng

Hình 18: The Users management layout in the admin side



The screenshot shows a web-based administration interface for a product database. On the left is a sidebar with icons for Thống kê (Analytics), Người dùng (Users), Đơn hàng (Orders), Sản phẩm (Products), and Đăng xuất (Logout). The main area has a header with the logo 'Fashion Shop' and a search bar labeled 'Tim kiem sản phẩm'. The right side is titled 'Sản phẩm' (Products) and displays a table with columns: MÃ SẢN PHẨM (Product ID), TÊN SẢN PHẨM (Product Name), HÌNH ẢNH (Image), SỐ LƯỢNG (Quantity), and GIÁ (Price). The table contains five entries:

MÃ SẢN PHẨM	TÊN SẢN PHẨM	HÌNH ẢNH	SỐ LƯỢNG	GIÁ
395	Túi đeo chéo dây kéo xéo đen TX006		36	375,000VNĐ
392	Áo Khoác Da Lộn Tròn AK023 Màu Nâu		36	645,000VNĐ
393	Áo Khoác Da Lộn Tròn AK023 Màu Xám Xanh		36	645,000VNĐ
391	Áo Khoác Da Lộn Tròn AK023 Màu Đen		36	645,000VNĐ

At the bottom of the table is a blue button labeled 'Xem thêm' (View more) and a link 'Danh sách các sản phẩm' (List of products).

Hình 19: The Products management layout in the admin side

This screenshot shows the same product management interface as Figure 19, but with a much larger number of products listed, indicating a scrollable view. The table structure is identical, displaying product IDs, names, images, quantities, and prices. The list includes products such as 'Áo Khoác Da Lộn Tròn AK023 Màu Đen', 'Áo Khoác Da Lộn Túi Đắp AK022 Màu Bò', 'Áo Khoác Da Lộn Túi Đắp AK022 Màu Rêu', 'Áo Khoác Da Lộn Túi Đắp AK022 Màu Xám', and 'Áo Khoác Dù Rã Ngang AK027 Màu Xám'. A vertical scrollbar is visible on the right side of the table.

Hình 20: The Products management layout when loading many products in the admin side



MÃ ĐƠN HÀNG	MÃ THANH TOÁN MOMO	NGƯỜI NHẬN	NGÀY ĐẶT HÀNG	SỐ ĐIỆN THOẠI	ĐỊA CHỈ GIAO HÀNG	TỔNG GIÁ TRỊ	TRANG THÁI
5	MOMO1685677583023	Đào Linh Dan	2023-06-02T03:46:22.989Z	0369872838	123, An Bình, Linh Xuân, Dĩ An	1,735,000VND	Thanh toán thành công
4	MOMO16856772389630	Đào Thị Hương Thảo	2023-06-02T03:19:49.589Z	12345	Địa chỉ giao hàng	645,000VND	Thanh toán thành công
3	MOMO1685636398577	Trần Thị Út Đôn	2023-06-01T16:19:58.216Z	0123456789	123 Mộc Châu, Sơn La	595,000VND	Chưa thanh toán
2	MOMO1685635992613	Nguyễn Thị Bưởi	2023-06-01T16:13:12.257Z	099949924	123/1234 Nguyễn Tri Phương, Quận 10, TP HCM	1,470,000VND	Thanh toán thất bại
1	MOMO1685635595283	Nguyễn Thị Mai Hương	2023-06-01T16:06:34.848Z	099922999	ABC, Thủ Đức, TP HCM	5,160,000VND	Thanh toán thành công

Hình 21: The Orders management layout in the admin side

## 5 Conclusion

E-commerce shopping system integrated with MOMO payment - the Fashion shop is a software developed and designed based on well-known platforms, technologies, and libraries that are easy to implement, maintain, and reuse in other disciplines. This is a commercial application developed at the same time as the country's industrialization and modernization, suitable for strong and user-friendly deployment. Besides, online payment is one of the best choices for the customer who has very little free time for shopping.

### 5.1 Short Retrospective

#### 5.1.1 Lesson learned

After completing this project, our team has completed and learned a lot of experiences that can apply in the future:

- Understand the whole process of software development life cycles and methodologies to manage the project during development time.
- Enhance the system analysis and design skill, which is very important to process a problem statement of the project into needed artifacts that solve them with the best outcome.
- Advanced in documentation with necessary diagrams and writing skills.
- Understand the momo payment flow and successfully integrate it into our system.



- Get familiar with new frameworks such as ReactJS and enhance our programming skills using NodeJS (ExpressJS), and SQL query for developing the web-based application.
- Understand the mySQL database management tools, and associated programming libraries like Jquery, ChartJs, Slide,... for saving time and cost.
- Solved many problems and internal conflicts together in the process of working on this project, from which we have gained many experiences in working in a group and working with partners.

### 5.1.2 Limitations in the project

Although our project has been successful to some extent, it still contains a lot of disadvantages that need to improve in the next phase in the future:

- The system still lacks several layouts for administration modules such as add or update a user.
- Our system also needs to enhance the security in the database like encrypting user passwords,..
- Even though we already implemented the comment API, due to sudden situations and time limitations, it still not be contained in the user interface. We will implement the layout to allow the user to comment on products via UI not via Postman anymore,...
- We also will implement the warranty policy that means when the customer buys goods within the warranty period but the goods have problems, can keep, and note the problem encountered to the warranty support staff

## 5.2 Conclusion

To sum up, our system is an e-commerce website that allows users to view/ search the available products and add desired products to their carts and can pay for a particular order via the Momo app in order to buy them. We also handle some functions on the admin side to allow the administrator can manage all entities in our system such as Products, Users, Statistics, etc.

Via this project, we have learned a lot of skills and knowledge about not only code skills but also system development methodologies and collaboration skills. All of them are valuable skills that can be transferred to serve our job in the near future as software developers.



## 6 References

1. *ReactJS*: <https://legacy.reactjs.org/>
2. *Learn ReactJS Tutorial*, JavaTpoint: <https://www.javatpoint.com/reactjs-tutorial>
3. *NodeJS*: [https://www.w3schools.com/nodejs/nodejs\\_intro.asp](https://www.w3schools.com/nodejs/nodejs_intro.asp)
4. *MySQL – Introduction*, Geeksforgeeks: <https://www.geeksforgeeks.org/mysql-introduction/>