

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO BÀI TẬP LỚN

Môn: Lập trình hướng đối tượng

Đề tài : Thiết kế Website bán giày

Nhóm lớp: 05

Nhóm 09

Thành Viên nhóm

Vũ Thành Long B21DCAT012

Trần Đức Khải B21DCAT108

Lê Quang Tuấn B21DCAT207

Mẫn Văn Trường B21DCCN120

Nguyễn Minh Nghĩa B21DCAT144

Giảng viên giảng dạy: Nguyễn Mạnh Sơn

Hà Nội – 2023

GIỚI THIỆU NHÓM VÀ PHÂN CHIA CÔNG VIỆC	4
THAM GIA ĐÓNG GÓP CODE WEB	5
Lý do chọn đề tài:.....	6
1. TỔNG QUAN	6
1.1. KHẢO SÁT THỰC TRẠNG	6
1.2. MỤC ĐÍCH VÀ MỤC TIÊU.....	7
1.2.1 Mục đích	7
1.2.2 Mục tiêu	7
1.3. ĐỐI TƯỢNG VÀ PHẠM VI CỦA ỨNG DỤNG	7
1.3.1 Đối tượng	7
1.3.2 Phạm vi	7
1.4. TÀI NGUYÊN	8
1.5. XÁC ĐỊNH YÊU CẦU CỦA KHÁCH HÀNG.....	8
1.5.1 Người bán.....	8
1.5.2 Khách hàng	8
2. MÔ HÌNH HÓA	8
2.1 SƠ ĐỒ KHỐI CHỨC NĂNG ỨNG DỤNG.....	8
2.2 CƠ SỞ DỮ LIỆU CỦA HỆ THỐNG	9
2.2.1 Bảng Account:	9
2.2.2 Bảng item:	9
2.2.3 Bảng order:.....	9
2.2.4 Bảng product.....	9
2.3.CÁC THÀNH PHẦN CHỨC NĂNG CỦA HỆ THỐNG.....	10
2.3.1. Kết nối đến cơ sở dữ liệu	10
2.3.2. Chức năng đăng nhập, đăng kí, đăng xuất trong hệ thống	10
2.3.3. Chức năng quản lý sản phẩm thêm , sửa, xóa sản phẩm.....	10
2.3.4. Chức năng quản lý giỏ hàng thêm , xóa sản phẩm.....	10
2.3.5. Chức năng quản lý đơn xóa đơn.	10
2.3.6. Hiển thị sản phẩm lên trang chủ và chi tiết sản phẩm.....	10
2.3.7. Chức năng tìm kiếm sản phẩm.....	10
2.4 SOURCE CODE CỦA CHƯƠNG TRÌNH	10
2.4.1. Source code DBContext.....	10
2.4.2. Source code DAO:	11
2.4.3. Source code Đăng nhập.....	16
2.4.4. Source code Đăng kí	17
2.4.5. Source code Đăng xuất	18
2.4.6. Source code Trang home:	18

2.4.7. Source code Chi tiết sản phẩm:.....	19
2.4.8. Source code Tìm kiếm sản phẩm:	19
2.4.9. Source code trang quản lý sản phẩm:.....	20
2.4.10. Source code thêm, sửa, xóa sản phẩm và load sản phẩm lên để sửa:.....	21
2.4.11. Source code trang quản lí đơn hàng	23
2.4.12. Source xóa các đơn hàng trong quản lí đơn hàng	24
2.4.13. Source code trang giỏ hàng	25
2.4.14. Source code xóa sản phẩm giỏ hàng	26
2.4.15. Source code thêm sản phẩm vào giỏ hàng	26
2.4.16. Source code mua hàng	27
2.4.17. Source code class Account:.....	28
2.4.18. Source code class Order:.....	29
2.4.19. Source code class Product:.....	30
2.5 THIẾT KẾ GIAO DIỆN HỆ THỐNG	30
2.5.1. Form Header	30
2.5.2. Form đăng nhập	31
2.5.3. Form Đăng kí	32
2.5.4. Form Chi tiết sản phẩm	33
2.5.5. Form Quản Lý Sản Phẩm.....	34
2.5.6. Form Trang Edit.....	34
2.5.7. Form Quản lý giỏ hàng	35
2.5.8. Form Quản lý đơn hàng	36
3. KẾT LUẬN.....	36
3.1. KẾT LUẬN ĐỀ TÀI	36
3.2. LỜI KẾT.....	37
Tài liệu tham khảo	37

GIỚI THIỆU NHÓM VÀ PHÂN CHIA CÔNG VIỆC

Sinh Viên	Nội Dung Tham Gia
Vũ Thành Long B21DCAT012 (Trưởng Nhóm)	<ul style="list-style-type: none"> • Phân chia công việc cho các thành viên • Thuyết trình • Tham gia lên ý tưởng cho nội dung • Tham gia đóng góp code cho ứng dụng • Tham gia thiết kế giao diện • Tham gia thiết kế cơ sở dữ liệu • Tìm tài liệu tham khảo • Viết báo cáo • Viết hướng dẫn chạy web
Trần Đức Khải B21DCAT108	<ul style="list-style-type: none"> • Tham gia lên ý tưởng cho nội dung • Tham gia đóng góp code cho ứng dụng • Tham gia thiết kế giao diện • Tìm tài liệu tham khảo • Viết báo cáo
Lê Quang Tuấn B21DCAT207	<ul style="list-style-type: none"> • Tham gia lên ý tưởng cho nội dung • Tham gia đóng góp code cho ứng dụng • Thiết kế cơ sở dữ liệu • Tìm tài liệu tham khảo • Viết báo cáo
Mẫn Văn Trường B21DCCN120	<ul style="list-style-type: none"> • Tham gia lên ý tưởng cho nội dung • Tham gia đóng góp code cho ứng dụng • Thiết kế cơ sở dữ liệu • Tìm tài liệu tham khảo • Viết báo cáo
Nguyễn Minh Nghĩa B21DCAT144	<ul style="list-style-type: none"> • Tham gia lên ý tưởng cho nội dung • Tham gia đóng góp code cho ứng dụng • Thư ký trong các buổi họp • Tìm tài liệu tham khảo • Viết báo cáo

THAM GIA ĐÓNG GÓP CODE WEB

Sinh Viên	Tham Gia Code
Vũ Thành Long B21DCAT012 (Trưởng Nhóm)	<ul style="list-style-type: none"> • BuyControl (Đặt hàng) • CartControl • AddControl (Thêm vào manager product) • DeleteControl (Xóa sản phẩm khỏi manager product) • EditControl (Sửa sản phẩm trong manager product) • LoadControl (Load sản phẩm lên để sửa) • SignupControl • LogoutControl • SearchControl • DAO và các class
Trần Đức Khải B21DCAT108	<ul style="list-style-type: none"> • LoginControl • AddCartControl • Tìm ảnh cho trang web
Lê Quang Tuấn B21DCAT207	<ul style="list-style-type: none"> • DBContext • DeleteCartControl • Dữ liệu mẫu thêm vào DB.
Mẫn Văn Trường B21DCCN120	<ul style="list-style-type: none"> • ManagerControl (Quản lí sản phẩm) • DeleteOrderControl • ManagerOrder
Nguyễn Minh Nghĩa B21DCAT144	<ul style="list-style-type: none"> • HomeController • DetailControl • Thông tin detail chi tiết sản phẩm

Lý do chọn đề tài:

- ❖ Trong những năm gần đây, việc mua sắm trực tuyến không chỉ là một xu hướng mà còn trở thành lựa chọn phổ biến của người tiêu dùng, đặc biệt trong lĩnh vực thời trang, bao gồm cả thời trang. Trong đó các trang web thương mại điện tử lớn như Shopee, Lazada, Tiki, ... đang dần cho thấy việc đơn giản hoá mua bán một cách nhanh chóng, an toàn và thuận tiện cho người tiêu dùng. Điều này tạo ra một cơ hội lớn cho việc xây dựng một nền tảng web thương mại điện tử chất lượng, thân thiện với người dùng và an toàn.
- ❖ Việc bán giày thông qua cửa hàng hay livestream đem lại hiệu quả không cao, tốn chi phí và không liệt kê hết tất cả thông tin cần biết về sản phẩm. Vì vậy web bán giày xuất hiện là tất yếu đáp ứng nhu cầu về tiện ích, dễ sử dụng và bảo mật cao.
- ❖ Số lượng khách hàng và hàng hoá rất lớn nên số lượng thông tin cần quản lý là rất lớn. Điều này sẽ gây khó khăn cho người mua cũng như bán hàng.
- ❖ Đề tài này rất thiết thực cho các nhà sản xuất và các cửa hàng bán giày dép đang hoạt động không chỉ riêng thị trường Việt Nam mà còn trên toàn cầu.
- ❖ Qua đề tài cũng là cách kiểm tra hiệu quả những kiến thức đã học và cũng là cách tiếp cận với thực tế có hiệu quả nhất đối với sinh viên.

Xuất phát từ những khó khăn và lợi ích đem lại nhóm em xin chọn đề tài ***“Website bán giày”***.

Với đề tài này, nhóm em mong muốn áp dụng các kiến thức đã được học trong trường cùng với việc tìm hiểu nghiên cứu ngôn ngữ và môi trường lập trình để xây dựng một web bán giày nhiều tiện ích và dễ sử dụng.

Nhóm 09 mong nhận được đóng góp ý kiến của thầy để đề tài được hoàn thiện hơn. Nhóm 09 xin chân thành cảm ơn!

1. TỔNG QUAN

1.1. KHẢO SÁT THỰC TRẠNG

- Hiện nay, với sự phát triển không ngừng của nền kinh tế, đòi hỏi mỗi doanh nghiệp phải có những chiến lược cụ thể phù hợp và đúng đắn để có thể giữ vững vị trí, cũng như cạnh tranh và phát triển. Một trong số các giải pháp được đưa ra ở đây là ứng dụng tiến bộ của công nghệ thông tin vào công việc bán hàng. Những công việc hàng ngày không còn là thao tác thủ công với những xếp tài liệu nữa mà thay vào đó là những chương trình hỗ trợ cho mọi công việc. Nó sẽ giảm thiểu được những khó khăn trong công việc bán hàng: nhân lực, thời gian, độ chính xác của thông tin, ... giúp cho công

ty, doanh nghiệp, cửa hàng có thể đưa ra những quyết định kinh doanh chính xác, đúng thời điểm cho việc kinh doanh trở nên hiệu quả hơn.

- Do đó dự án xây dựng và phát triển một trang web bán hàng ứng dụng kỹ thuật, công nghệ mới nhất giúp cho việc quảng bá sản phẩm của doanh nghiệp, công ty, cửa hàng được mọi người biết đến rộng rãi trên toàn quốc.

1.2. MỤC ĐÍCH VÀ MỤC TIÊU

1.2.1 Mục đích

- Xây dựng một website bán hàng qua mạng

1.2.2 Mục tiêu

- Các chức năng cơ bản đề ra ban đầu:

- Đăng nhập, đăng kí.
- Quản lí sản phẩm bán, giỏ hàng, quản lí đơn hàng
- Tìm kiếm sản phẩm trên trang
- Thông tin chi tiết sản phẩm
- Thông báo trên trang web mỗi khi sale tới
- Phân trang và có Category

- Các chức năng đã đạt được:

- Đăng nhập, đăng kí.
- Quản lí sản phẩm bán, giỏ hàng, quản lí đơn hàng
- Tìm kiếm sản phẩm trên trang
- Thông tin chi tiết sản phẩm

- Đáp ứng được tiêu chuẩn của một website:

- Cấu trúc khoa học, chặt chẽ, rõ ràng, dễ dàng điều hướng, khai thác, chỉnh sửa.
- Thân thiện với người sử dụng.
- Nội dung được tối ưu, tốc độ tải/ nạp trang web nhanh.
- Hoạt động ổn định.

- Đáp ứng các chức năng khách hàng yêu cầu (quảng bá sản phẩm và bán hàng trực tiếp qua mạng)

1.3. ĐỐI TƯỢNG VÀ PHẠM VI CỦA ỨNG DỤNG

1.3.1 Đối tượng

Hệ thống quản lý website bán giày được xây dựng hướng đến các đối tượng:

- Người quản trị hệ thống
- Khách hàng

1.3.2 Phạm vi

- Dự án xây dựng website quảng bá sản phẩm và bán sản phẩm trực tuyến cho khách hàng thông qua hệ thống Internet.

- Website được sử dụng cho các cửa hàng bán giày và các khách hàng muốn mua hàng trên Website.

1.4. TÀI NGUYÊN

- Các cửa hàng cung cấp:
 - + Cung cấp chi phí dự án.
 - + Nghiệp vụ quản lý
 - + Thông tin sản phẩm
 - + Các mẫu văn bản cần thiết.
- ...
- Khách hàng mua giày:
 - + Cung cấp các nhu cầu, mong muốn khi mua hàng.
 - + Các thông tin cần thiết để mua hàng
- ...

1.5. XÁC ĐỊNH YÊU CẦU CỦA KHÁCH HÀNG

1.5.1 Người bán

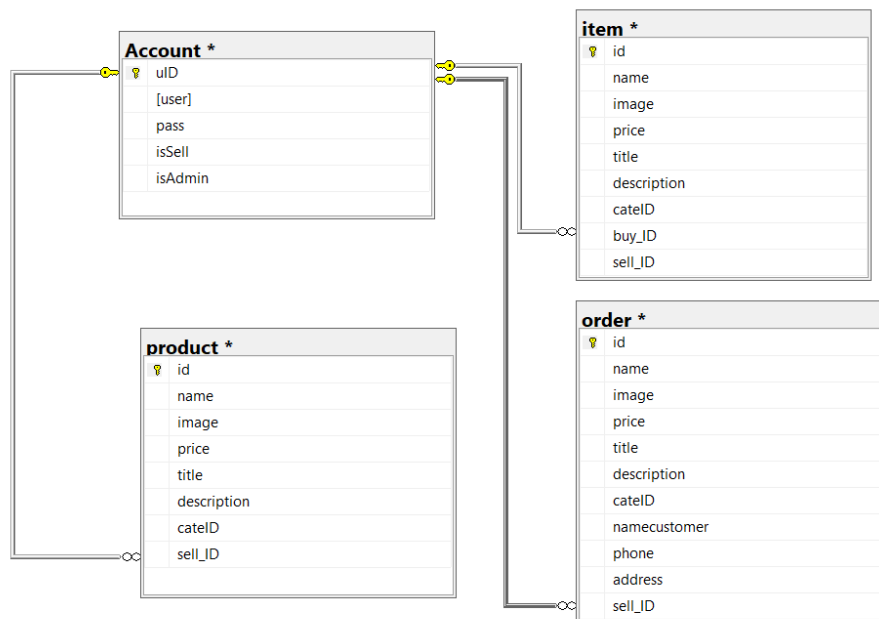
- Quản lý sản phẩm.
- Thông tin chi tiết sản phẩm được tải lên hệ thống
- Quản lý thông tin đơn mua.

1.5.2 Khách hàng

- Xem và mua hàng trực tuyến qua mạng.
- Đăng ký thành viên.
- Tìm kiếm sản phẩm mà khách hàng quan tâm.
- Giỏ hàng để lưu những sản phẩm khách hàng muốn mua.

2. MÔ HÌNH HÓA

2.1 SƠ ĐỒ KHỐI CHỨC NĂNG ỨNG DỤNG



2.2 CƠ SỞ DỮ LIỆU CỦA HỆ THỐNG

Để nắm được yêu cầu của bài toán, chúng ta cần hiểu rõ về cơ sở dữ liệu mà bài toán cần. Ở đây nhóm 09 xin đưa ra những thông tin mà phần mềm web bán hàng có như sau:

2.2.1 Bảng Account:

Column Name	Data Type	Allow Nulls
uID	int	<input type="checkbox"/>
[user]	varchar(255)	<input checked="" type="checkbox"/>
pass	varchar(255)	<input checked="" type="checkbox"/>
isSell	int	<input checked="" type="checkbox"/>
isAdmin	int	<input checked="" type="checkbox"/>

2.2.2 Bảng item:

Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
name	nvarchar(MAX)	<input checked="" type="checkbox"/>
image	nvarchar(MAX)	<input checked="" type="checkbox"/>
price	money	<input checked="" type="checkbox"/>
title	nvarchar(MAX)	<input checked="" type="checkbox"/>
description	nvarchar(MAX)	<input checked="" type="checkbox"/>
catelD	int	<input checked="" type="checkbox"/>
buy_ID	int	<input checked="" type="checkbox"/>
sell_ID	int	<input checked="" type="checkbox"/>

2.2.3 Bảng order:

Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
name	nvarchar(MAX)	<input checked="" type="checkbox"/>
image	nvarchar(MAX)	<input checked="" type="checkbox"/>
price	money	<input checked="" type="checkbox"/>
title	nvarchar(MAX)	<input checked="" type="checkbox"/>
description	nvarchar(MAX)	<input checked="" type="checkbox"/>
catelD	int	<input checked="" type="checkbox"/>
namecustomer	nvarchar(MAX)	<input checked="" type="checkbox"/>
phone	nvarchar(MAX)	<input checked="" type="checkbox"/>
address	nvarchar(MAX)	<input checked="" type="checkbox"/>
sell_ID	int	<input checked="" type="checkbox"/>

2.2.4 Bảng product

Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
name	nvarchar(MAX)	<input checked="" type="checkbox"/>
image	nvarchar(MAX)	<input checked="" type="checkbox"/>
price	money	<input checked="" type="checkbox"/>
title	nvarchar(MAX)	<input checked="" type="checkbox"/>
description	nvarchar(MAX)	<input checked="" type="checkbox"/>
catelD	int	<input checked="" type="checkbox"/>
sell_ID	int	<input checked="" type="checkbox"/>

2.3.CÁC THÀNH PHẦN CHỨC NĂNG CỦA HỆ THỐNG

2.3.1. Kết nối đến cơ sở dữ liệu

- Kết nối đến cơ sở dữ liệu để lấy thông tin.

2.3.2. Chức năng đăng nhập, đăng kí, đăng xuất trong hệ thống

- Đăng nhập tài khoản để có thể tìm kiếm, mua hàng và quản lý sản phẩm, quản lý đơn hàng.
- Đăng kí để có tài khoản
- Đăng xuất để thoát phiên đăng nhập trong hệ thống.

2.3.3. Chức năng quản lý sản phẩm thêm , sửa, xóa sản phẩm.

- Thêm sản phẩm để hiển thị sản phẩm lên trang.
- Sửa để chỉnh sửa thông tin sản phẩm.
- Xóa để loại sản phẩm ra khỏi hệ thống bán.

2.3.4. Chức năng quản lý giỏ hàng thêm , xóa sản phẩm.

- Thêm sản phẩm quan tâm vào giỏ đợi thanh toán.
- Xóa để loại sản phẩm ra khỏi hệ thống bán.

2.3.5. Chức năng quản lý đơn xóa đơn.

- Xóa những đơn hàng đã thực hiện thành công ra khỏi hệ thống.

2.3.6. Hiển thị sản phẩm lên trang chủ và chi tiết sản phẩm.

- Hiển thị sản phẩm mà người bán cung cấp lên trang chủ.
- Chi tiết mô tả sản phẩm để người quan tâm có nhiều thông tin hơn.

2.3.7. Chức năng tìm kiếm sản phẩm.

- Tìm kiếm để tìm nhanh sản phẩm ta quan tâm đến.

2.4 SOURCE CODE CỦA CHƯƠNG TRÌNH

2.4.1. Source code DBContext

- ❖ **Chức năng : Dùng để kết nối cơ sở dữ liệu MySql và NetBeans**

```

package context;

import java.sql.Connection;
import java.sql.DriverManager;

public class DBContext {

    /*USE BELOW METHOD FOR YOUR DATABASE CONNECTION FOR BOTH SINGLE AND MULTIPLE SQL SERVER INSTANCE(s)*/
    /*DO NOT EDIT THE BELOW METHOD, YOU MUST USE ONLY THIS ONE FOR YOUR DATABASE CONNECTION*/
    public Connection getConnection() throws Exception {
        String url = "jdbc:sqlserver://" + serverName + ":" + portNumber + "\\\\" + instance + ";databaseName=" + dbName;
        if(instance == null || instance.trim().isEmpty())
            url = "jdbc:sqlserver://" + serverName + ":" + portNumber + ";databaseName=" + dbName;
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        return DriverManager.getConnection(url, userID, password);
    }
    /*Insert your other code right after this comment*/
    /*Change/update information of your database connection, DO NOT change name of instance variables in this class*/
    private final String serverName = "LONGMOI\\SQLEXPRESS";
    private final String dbName = "Wish";
    private final String portNumber = "1433";
    private final String instance = ""; //LEAVE THIS ONE EMPTY IF YOUR SQL IS A SINGLE INSTANCE
    private final String userID = "longmoi";
    private final String password = "long0965813502";
    public static void main(String[] args) {
        try {
            System.out.println(new DBContext().getConnection());
        } catch (Exception e) {
        }
    }
}

```

2.4.2. Source code DAO:

❖ Chức năng : Load dữ liệu từ DB lên phục vụ cho các chức năng bên dưới

```

2 package dao;
3
4 import context.DBContext;
5 import entity.*;
6 import java.sql.Connection;
7 import java.sql.PreparedStatement;
8 import java.sql.ResultSet;
9 import java.time.LocalDate;
10 import java.util.ArrayList;
11 import java.util.List;
12
13 public class DAO {
14
15     Connection conn = null;
16     PreparedStatement ps = null;
17     ResultSet rs = null;
18
19     public List<Product> getAllProduct() {
20         List<Product> list = new ArrayList<>();
21         String query = "select * from product";
22         try {
23             conn = new DBContext().getConnection(); //mo ket noi voi sql
24             ps = conn.prepareStatement(query);
25             rs = ps.executeQuery();
26             while (rs.next()) {
27                 list.add(new Product(rs.getInt(1),
28                                     rs.getString(2),
29                                     rs.getString(3),
30                                     rs.getDouble(4),
31                                     rs.getString(5),
32                                     rs.getString(6)));
33             }
34         } catch (Exception e) {
35         }
36         return list;
37     }
38
39     public List<Product> getProductByCID(String cid) {
40         List<Product> list = new ArrayList<>();

```

```

40 List<Product> list = new ArrayList<>();
41 String query = "select * from product\n"
42     + "where cateID = ?";
43 try {
44     conn = new DBContext().getConnection();//mo ket noi voi sql
45     ps = conn.prepareStatement(query);
46     ps.setString(1, cid);
47     rs = ps.executeQuery();
48     while (rs.next()) {
49         list.add(new Product(rs.getInt(1),
50             rs.getString(2),
51             rs.getString(3),
52             rs.getDouble(4),
53             rs.getString(5),
54             rs.getString(6)));
55     }
56 } catch (Exception e) {
57 }
58 return list;
59 }
60
61 public List<Product> getProductBySellID(int id) {
62     List<Product> list = new ArrayList<>();
63     String query = "select * from product\n"
64         + "where sell_ID = ?";
65     try {
66         conn = new DBContext().getConnection();//mo ket noi voi sql
67         ps = conn.prepareStatement(query);
68         ps.setInt(1, id);
69         rs = ps.executeQuery();
70         while (rs.next()) {
71             list.add(new Product(rs.getInt(1),
72                 rs.getString(2),
73                 rs.getString(3),
74                 rs.getDouble(4),
75                 rs.getString(5),
76                 rs.getString(6)));
77         }
78     } catch (Exception e) {
79     }

```

```

79     }
80     return list;
81 }
82
83 public List<Product> getProductByBuyID(int id) {
84     List<Product> list = new ArrayList<>();
85     String query = "select * from item\n"
86         + "where buy_ID = ?";
87     try {
88         conn = new DBContext().getConnection();//mo ket noi voi sql
89         ps = conn.prepareStatement(query);
90         ps.setInt(1, id);
91         rs = ps.executeQuery();
92         while (rs.next()) {
93             list.add(new Product(rs.getInt(1),
94                 rs.getString(2),
95                 rs.getString(3),
96                 rs.getDouble(4),
97                 rs.getString(5),
98                 rs.getString(6)));
99         }
100     } catch (Exception e) {
101     }
102     return list;
103 }
104
105 public List<Order> getOrderBySellID(int sellID) {
106     List<Order> list = new ArrayList<>();
107     String query = "select * from [dbo].[order]\n"
108         + "where sell_ID = ?";
109     try {
110         conn = new DBContext().getConnection();//mo ket noi voi sql
111         ps = conn.prepareStatement(query);
112         ps.setInt(1, sellID);
113         rs = ps.executeQuery();
114         while (rs.next()) {
115             list.add(new Order(rs.getInt(1),
116                 rs.getString(2),
117                 rs.getString(3),
118                 rs.getDouble(4),

```

```

118         rs.getDouble(4),
119         rs.getString(5),
120         rs.getString(6),
121         rs.getString(8),
122         rs.getString(9),
123         rs.getString(10));
124     }
125     } catch (Exception e) {
126     }
127     return list;
128 }
129
130 public List<Product> searchByName(String txtSearch) {
131     List<Product> list = new ArrayList<>();
132     String query = "select * from product "
133         + "where [name] like '%" + txtSearch + "%'";
134     try {
135         conn = new DBContext().getConnection();//mo ket noi voi sql
136         ps = conn.prepareStatement(query);
137         rs = ps.executeQuery();
138         while (rs.next()) {
139             list.add(new Product(rs.getInt(1),
140                 rs.getString(2),
141                 rs.getString(3),
142                 rs.getDouble(4),
143                 rs.getString(5),
144                 rs.getString(6)));
145         }
146     } catch (Exception e) {
147     }
148     return list;
149 }
150
151 public Product getProductByID(String id) {
152     String query = "select * from product\n"
153         + "where id = ?";
154     try {
155         conn = new DBContext().getConnection();//mo ket noi voi sql
156         ps = conn.prepareStatement(query);
157         ps.setString(1, id);
158
159         rs = ps.executeQuery();
160         while (rs.next()) {
161             return new Product(rs.getInt(1),
162                 rs.getString(2),
163                 rs.getString(3),
164                 rs.getDouble(4),
165                 rs.getString(5),
166                 rs.getString(6));
167         }
168     } catch (Exception e) {
169     }
170     return null;
171 }
172
173 public int getSellIDByProductID(String id) {
174     String query = "select * from product\n"
175         + "where id = ?";
176     try {
177         conn = new DBContext().getConnection();//mo ket noi voi sql
178         ps = conn.prepareStatement(query);
179         ps.setString(1, id);
180         rs = ps.executeQuery();
181         while (rs.next()) {
182             return rs.getInt(8);
183         }
184     } catch (Exception e) {
185     }
186     return 0;
187 }
188
189 public int getSellIDByItemID(String id) {
190     String query = "select * from item\n"
191         + "where id = ?";
192     try {
193         conn = new DBContext().getConnection();//mo ket noi voi sql
194         ps = conn.prepareStatement(query);
195         ps.setString(1, id);
196         rs = ps.executeQuery();
197         while (rs.next()) {

```

```

196         while (rs.next()) {
197             return rs.getInt(9);
198         }
199     } catch (Exception e) {
200     }
201     return 0;
202 }
203
204
205 public Account login(String user, String pass){
206     String query ="select *from account\n"
207         +"where [user] =?\n"
208         +"and pass = ?";
209     try {
210         conn = new DBContext().getConnection();//mo ket noi voi sql
211         ps = conn.prepareStatement(query);
212         ps.setString(1, user);
213         ps.setString(2, pass);
214         rs = ps.executeQuery();
215         while(rs.next()){
216             return new Account(rs.getInt(1),
217                 rs.getString(2),
218                 rs.getString(3),
219                 rs.getInt(4),
220                 rs.getInt(5));
221         }
222     } catch (Exception e) {
223     }
224     return null;
225 }
226
227 public Account checkAccountExist(String user){
228     String query ="select *from account\n"
229         +"where [user] =?\n";
230     try {
231         conn = new DBContext().getConnection();//mo ket noi voi sql
232         ps = conn.prepareStatement(query);
233         ps.setString(1, user);
234         rs = ps.executeQuery();
235         while(rs.next()){

```

```

235         while(rs.next()){
236             return new Account(rs.getInt(1),
237                 rs.getString(2),
238                 rs.getString(3),
239                 rs.getInt(4),
240                 rs.getInt(5));
241         }
242     } catch (Exception e) {
243     }
244     return null;
245 }
246
247 public void signup(String user, String pass){
248     String query="insert into account\n"+"values(?,?,0,0)";
249     try {
250         conn = new DBContext().getConnection();//mo ket noi voi sql
251         ps = conn.prepareStatement(query);
252         ps.setString(1, user);
253         ps.setString(2, pass);
254         ps.executeUpdate();
255     } catch (Exception e) {
256     }
257 }
258 public void deleteProduct(String pid){
259     String query="delete from product\n"
260         +"where id = ?";
261     try {
262         conn = new DBContext().getConnection();//mo ket noi voi sql
263         ps = conn.prepareStatement(query);
264         ps.setString(1, pid);
265         ps.executeUpdate();
266     } catch (Exception e) {
267     }
268 }
269 public void insertProduct(String name, String image, String price,
270     String title, String description, String category, int sid) {
271     String query = "INSERT [dbo].[product] \n"
272         + "([name], [image], [price], [title], [description], [cateID], [sell_ID])\n"
273         + "VALUES(?,?,?,?,?,?)";
274     try {

```

```

274     try {
275         conn = new DBContext().getConnection();//mo ket noi voi sql
276         ps = conn.prepareStatement(query);
277         ps.setString(1, name);
278         ps.setString(2, image);
279         ps.setString(3, price);
280         ps.setString(4, title);
281         ps.setString(5, description);
282         ps.setString(6, category);
283         ps.setInt(7, sid);
284         ps.executeUpdate();
285     } catch (Exception e) {
286     }
287 }
288
289 public void insertItem(String name, String image, String price,
290     String title, String description, String category, int bid,int sellID) {
291     String query = "INSERT [dbo].[item] \n"
292         + "([name], [image], [price], [title], [description], [cateID], [buy_ID],[sell_ID])\n"
293         + "VALUES(?, ?, ?, ?, ?, ?, ?, ?)";
294     try {
295         conn = new DBContext().getConnection();//mo ket noi voi sql
296         ps = conn.prepareStatement(query);
297         ps.setString(1, name);
298         ps.setString(2, image);
299         ps.setString(3, price);
300         ps.setString(4, title);
301         ps.setString(5, description);
302         ps.setString(6, category);
303         ps.setInt(7, bid);
304         ps.setInt(8, sellID);
305         ps.executeUpdate();
306     } catch (Exception e) {
307     }
308 }
309
310
311 public void insertOrder(String name, String image, String price,
312     String title, String description, String category, String namecustomer, String phone, String address, int sellID) {
313     String query = "INSERT [dbo].[order] \n"
314         + "([name], [image], [price], [title], [description], [cateID], [namecustomer], [phone], [address],[sell_ID])\n"
315         + "VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?)";
316     try {
317         conn = new DBContext().getConnection();//mo ket noi voi sql
318         ps = conn.prepareStatement(query);
319         ps.setString(1, name);
320         ps.setString(2, image);
321         ps.setString(3, price);
322         ps.setString(4, title);
323         ps.setString(5, description);
324         ps.setString(6, category);
325         ps.setString(7, namecustomer);
326         ps.setString(8, phone);
327         ps.setString(9, address);
328         ps.setInt(10, sellID);
329         ps.executeUpdate();
330     } catch (Exception e) {
331     }
332 }
333
334 public void editProduct(String name, String image, String price,
335     String title, String description, String category, String pid) {
336     String query = "update product\n"
337         + "set [name] = ?,\n"
338         + "[image] = ?,\n"
339         + "[price] = ?,\n"
340         + "[title] = ?,\n"
341         + "[description] = ?,\n"
342         + "[cateID] = ?\n"
343         + "where id = ?";
344     try {
345         conn = new DBContext().getConnection();//mo ket noi voi sql
346         ps = conn.prepareStatement(query);
347         ps.setString(1, name);
348         ps.setString(2, image);
349         ps.setString(3, price);
350         ps.setString(4, title);
351         ps.setString(5, description);
352         ps.setString(6, category);

```

```

352         ps.setString(6, category);
353         ps.setString(7, pid);
354         ps.executeUpdate();
355     } catch (Exception e) {
356     }
357 }
358
359 public void deleteItem(String pid){
360     String query="delete from item\n"
361         +"where id = ?";
362     try {
363         conn = new DBContext().getConnection();//mo ket noi voi sql
364         ps = conn.prepareStatement(query);
365         ps.setString(1, pid);
366         ps.executeUpdate();
367     } catch (Exception e) {
368     }
369 }
370
371 public void deleteOrder(String pid){
372     String query="delete from [dbo].[order]\n"
373         +"where id = ?";
374     try {
375         conn = new DBContext().getConnection();//mo ket noi voi sql
376         ps = conn.prepareStatement(query);
377         ps.setString(1, pid);
378         ps.executeUpdate();
379     } catch (Exception e) {
380     }
381 }
382
383 public static void main(String[] args) {
384 }
385
386 }
387

```

2.4.3. Source code Đăng nhập

❖ Chức năng : Dùng để đăng nhập vào hệ thống

```

package control;

import dao.DAO;
import entity.Account;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;

@WebServlet(name="LoginControl", urlPatterns={"/login"})
public class LoginControl extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String username = request.getParameter("user");
        String password = request.getParameter("pass");
        DAO dao = new DAO();
        Account a = dao.login(username, password);
        if(a==null){
            request.setAttribute("mess", "Wrong Username or Password");
            request.getRequestDispatcher("login.jsp").forward(request, response);
        }else{
            HttpSession session = request.getSession();
            session.setAttribute("acc", a);
            // session.setMaxInactiveInterval(10); //thoi gian cho session ton tai
            response.sendRedirect("home");
        }
    }

    HttpServlet methods. Click on the + sign on the left to edit the code.
}

```


- Thực hiện lấy user và pass từ phần text được nhập ở giao diện bằng `getParameter`.
- Sử dụng hàm `dao.login` để so sánh trong hệ thống nếu không có thì sẽ báo Wrong và chuyển hướng quay lại trang login.
- Có thì ta thực hiện đăng nhập và chuyển hướng tới trang home.

2.4.4. Source code Đăng kí

❖ Chức năng : Dùng để đăng kí hệ thống

```
package control;

import dao.DAO;
import entity.Account;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

@WebServlet(name="SignUpControll", urlPatterns={"/signup"})
public class SignUpControll extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String user = request.getParameter("user");
        String pass = request.getParameter("pass");
        String re_pass = request.getParameter("repass");
        if(!pass.equals(re_pass)){
            request.getRequestDispatcher("register.jsp").forward(request, response);
        }
        else{
            DAO dao = new DAO();
            Account a = dao.checkAccountExist(user);
            if(a==null){
                dao.signup(user, pass);
                Account tmp = dao.login(user, pass);
                request.setAttribute("account", tmp);
                request.getRequestDispatcher("index.jsp").forward(request, response);
            }else{
                request.getRequestDispatcher("register.jsp").forward(request, response);
            }
        }
    }
}
```

HttpServlet methods. Click on the + sign on the left to edit the code.

- Thực hiện lấy user, pass, và repass từ phần text được nhập ở giao diện bằng `getParameter`.
- Đầu tiên nếu pass và repass không trùng thì ta quay lại trang đăng kí.
- Đã trùng rồi thì ta xét tài khoản đã tồn tại chưa bằng hàm `dao.checkAccountExist`:

- Nếu chưa có thì ta thực hiện đăng kí rồi chuyển đến trang home (`index.jsp`).
- Nếu có thì ta quay lại thực hiện đăng kí.

2.4.5. Source code Đăng xuất

❖ Chức năng : Dừng để đăng xuất khỏi hệ thống

```
package control;

import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;

@WebServlet(name="LogoutControl", urlPatterns={"/logout"})
public class LogoutControl extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        HttpSession session = request.getSession();
        session.removeAttribute("acc");
        response.sendRedirect("home");
    }

    HttpServlet methods. Click on the + sign on the left to edit the code.
}
```

- Thực hiện loại bỏ phiên và quay về trang chủ với trạng thái không dùng tài khoản.

2.4.6. Source code Trang home:

❖ Chức năng : Dừng để làm trang chủ chính trưng bày sản phẩm

```
package control;

import dao.DAO;
import entity.Product;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import static java.util.Collections.list;

@WebServlet(name = "HomeController", urlPatterns = {"/home"})
public class HomeController extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        //b1: get data from dao
        DAO dao = new DAO();
        List<Product> list = dao.getAllProduct();
        //b2: set data to jsp
        request.setAttribute("listP", list);
        request.getRequestDispatcher("index.jsp").forward(request, response);
    }

    HttpServlet methods. Click on the + sign on the left to edit the code.
}
```

- Tạo một list và lấy hết sản phẩm từ cơ sở dữ liệu bằng hàm dao.getAllProduct.

- setAttribute cho list là listP sử dụng để hiển thị các sản phẩm trong giao diện và chuyển hướng đến trang home (index.jsp).

2.4.7. Source code Chi tiết sản phẩm:

❖ Chức năng : Giới thiệu sản phẩm bán hàng

```
package control;

import dao.DAO;
import entity.Product;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

@WebServlet(name = "DetailControl", urlPatterns = {"/detail"})
public class DetailControl extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String id = request.getParameter("pid");
        DAO dao = new DAO();
        Product p = dao.getProductByID(id);

        request.setAttribute("detail", p);
        request.getRequestDispatcher("details.jsp").forward(request, response);
    }

    HttpServlet methods. Click on the + sign on the left to edit the code.
}
```

- Thực hiện lấy id của sản phẩm bằng getParameter.
- Từ id ta lấy sản phẩm bằng dao.getProductByID.
- setAttribute cho product là detail để có thể thực hiện hiển thị thông tin sản phẩm lên trang detail và chuyển hướng đến đó.

2.4.8. Source code Tìm kiếm sản phẩm:

❖ Chức năng : Tìm kiếm sản phẩm bán hàng

```

package control;

import dao.DAO;
import entity.Product;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

@WebServlet(name = "SearchControl", urlPatterns = {"/search"})
public class SearchControl extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        request.setCharacterEncoding("UTF-8");
        String txtSearch = request.getParameter("txt");//giay chay bo
        DAO dao = new DAO();
        List<Product> list = dao.searchByName(txtSearch);

        request.setAttribute("listP", list);
        request.setAttribute("txtS", txtSearch);
        request.getRequestDispatcher("index_1.jsp").forward(request, response);
    }

    HttpServlet methods. Click on the + sign on the left to edit the code.
}

```

- Lấy text nhập từ txtSearch bằng getParameter.
- Lấy list sản phẩm bằng dao.searchByName.
- setAttribute cho list là listP để hiển thị thông tin trên giao diện và chuyển hướng đến trang home sau khi thực hiện search (index_1.jsp).

2.4.9. Source code trang quản lý sản phẩm:

❖ **Chức năng : hiển thị sản phẩm bán.**

```

package control;

import dao.DAO;
import entity.Account;
import entity.Product;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import java.util.List;

@WebServlet(name="ManagerControl", urlPatterns={"/manager"})
public class ManagerControl extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        HttpSession session = request.getSession();
        Account a = (Account) session.getAttribute("acc");
        int id = a.getId();
        DAO dao = new DAO();
        List<Product> list = dao.getProductBySellID(id);

        request.setAttribute("listP", list);
        request.getRequestDispatcher("ManagerProduct.jsp").forward(request, response);
    }
}

```

HttpServlet methods. Click on the + sign on the left to edit the code.

- Lấy Account hiện tại bằng session.getAttribute("acc").
- Thực hiện lấy list sản phẩm bán thông qua id của Account bằng dao.getProductBySellID.
- setAttribute cho list là listP để hiển thị thông tin trên giao diện trang ManagerProduct và chuyển hướng đó.

2.4.10. Source code thêm, sửa, xóa sản phẩm và load sản phẩm lên để sửa:

❖ **Chức năng : Thêm sản phẩm bán**

```

package control;

import dao.DAO;
import entity.Account;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;

@WebServlet(name="AddControl", urlPatterns={"/add"})
public class AddControl extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String pname = request.getParameter("name");
        String pimage = request.getParameter("image");
        String pprice = request.getParameter("price");
        String ptitle = request.getParameter("title");
        String pdescription = request.getParameter("description");
        String pcategory = "1";
        HttpSession session = request.getSession();
        Account a = (Account) session.getAttribute("acc");
        int sid = a.getId();

        DAO dao = new DAO();
        dao.insertProduct(pname, pimage, pprice, ptitle, pdescription, pcategory, sid);
        response.sendRedirect("manager");
    }

    HttpServlet methods. Click on the + sign on the left to edit the code.
}

```

- Thực hiện get.Paramter để lấy thông tin cần thiết đồng thời lấy account hiện tại bằng session.getAttribute để sau đó lấy id acc.
- Thêm sản phẩm vào cơ sở dữ liệu để hiển thị lên giao diện bằng insertProduct và chuyển hướng đến ManagerProduct.

❖ Chức năng : Sửa sản phẩm bán

```

import dao.DAO;
import entity.Account;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;

@WebServlet(name = "EditControl", urlPatterns = {"/edit"})
public class EditControl extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        request.setCharacterEncoding("UTF-8");
        String pid = request.getParameter("id");
        String pname = request.getParameter("name");
        String pimage = request.getParameter("image");
        String pprice = request.getParameter("price");
        String ptitle = request.getParameter("title");
        String pdescription = request.getParameter("description");
        String pcategory = request.getParameter("category");
        DAO dao = new DAO();
        dao.editProduct(pname, pimage, pprice, ptitle, pdescription, pcategory, pid);
        response.sendRedirect("manager");
    }

    HttpServlet methods. Click on the + sign on the left to edit the code.
}

```

- Lấy text bằng `getParameter` và thực hiện lưu thông tin sửa bằng `dao.editProduct` và chuyển hướng đến trang `ManagerProduct`.

❖ Chức năng : Xóa sản phẩm bán

```
package control;

import dao.DAO;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

@WebServlet(name="DeleteControl", urlPatterns={"/delete"})
public class DeleteControl extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String pid = request.getParameter("pid");
        DAO dao = new DAO();
        dao.deleteProduct(pid);
        response.sendRedirect("manager");
    }

    HttpServlet methods. Click on the + sign on the left to edit the code.
}
```

- Lấy id của sản phẩm bằng `getParameter` và xóa sản phẩm trong cơ sở dữ liệu bằng `dao.deleteProduct` và chuyển hướng đến `ManagerProduct`.

❖ Chức năng : Load sản phẩm lên để sửa

```
package control;

import dao.DAO;
import entity.Product;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

@WebServlet(name = "LoadControl", urlPatterns = {"/loadProduct"})
public class LoadControl extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String id = request.getParameter("pid");
        DAO dao = new DAO();
        Product p = dao.getProductByID(id);

        request.setAttribute("detail", p);
        request.getRequestDispatcher("Edit.jsp").forward(request, response);
    }

    HttpServlet methods. Click on the + sign on the left to edit the code.
}
```

- Lấy id của sản phẩm bằng `getParameter` và lấy sản phẩm trong cơ sở dữ liệu bằng `dao.getProductByID` và chuyển hướng đến trang edit.

2.4.11. Source code trang quản lý đơn hàng

❖ Chức năng : Hiển thị đơn hàng được đặt

```

package control;

import dao.DAO;
import entity.Account;
import entity.Order;
import entity.Product;
import java.io.*;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import java.util.*;

@WebServlet(name="ManagerOrder", urlPatterns={"/order"})
public class ManagerOrder extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        HttpSession session = request.getSession();
        Account a = (Account) session.getAttribute("acc");
        int sellID = a.getId();
        DAO dao = new DAO();
        List<Order> list = dao.getOrderById(sellID);
        list.sort(new Comparator<Order>() {
            @Override
            public int compare(Order o1, Order o2) {
                return o1.getNamecustomer().compareTo(o2.getNamecustomer());
            }
        });

        request.setAttribute("listO", list);
        request.getRequestDispatcher("ManagerOrder.jsp").forward(request, response);
    }
}

```

HttpServlet methods. Click on the + sign on the left to edit the code.

- Dùng session.getAttribute để lấy Account hiện tại từ session để sau đó có thể lấy id của người bán.
- Lấy list order bằng hàm dao.getOrderById và thực hiện sắp xếp các Order theo tên của khách.
- setAttribute cho list order là listO để hiển thị danh sách đơn đặt lên giao diện và chuyển hướng đến trang ManagerOrder.

2.4.12. Source xóa các đơn hàng trong quản lý đơn hàng

❖ Chức năng : Xóa các đơn hàng trong danh sách đơn


```

package control;

import dao.DAO;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

@WebServlet(name="DeleteOrderControl", urlPatterns={"/deleteorder"})
public class DeleteOrderControl extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String pid = request.getParameter("pid");
        DAO dao = new DAO();
        dao.deleteOrder(pid);
        response.sendRedirect("order");
    }

    HttpServlet methods. Click on the + sign on the left to edit the code.
}

```

- Thực hiện lấy id của sản phẩm trong danh sách order bằng getParameter.
- Xóa đơn bằng lời gọi hàm dao.deleteOrder và truyền id của sản phẩm trong danh sách order vào.

2.4.13. Source code trang giỏ hàng

❖ Chức năng : Hiển thị sản phẩm khách muốn mua

```

package control;

import dao.DAO;
import entity.*;
import java.io.*;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import java.util.*;

@WebServlet(name="CartControl", urlPatterns={"/cart"})
public class CartControl extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        HttpSession session = request.getSession();
        Account a = (Account) session.getAttribute("acc");
        int id = a.getId();
        DAO dao = new DAO();
        List<Product> list = dao.getProductByBuyID(id);
        list.sort(new Comparator<Product>() {
            @Override
            public int compare(Product o1, Product o2) {
                return o1.getImage().compareTo(o2.getImage());
            }
        });
        long totalmoney = 0;
        for(Product i:list){totalmoney += i.getPrice();}
        request.setAttribute("totalmoney", totalmoney);
        request.setAttribute("listI", list);
        request.getRequestDispatcher("Cart.jsp").forward(request, response);
    }

    HttpServlet methods. Click on the + sign on the left to edit the code.
}

```

- Dùng session lấy Account hiện tại để sau đó lấy id người mua.
- Lấy list sản phẩm khách muốn mua thông qua dao.getProductByBuyID đồng thời sắp xếp theo ảnh (mục đích là để các sản phẩm giống nhau sẽ nằm chung 1 cụm).
- Thực hiện tính tổng tiền thông qua biến “totalmoney”.
- setAttribute cho list là listI và totalmoney là totalmoney để dễ dàng hiển thị lên giao diện giỏ hàng và chuyển hướng tới trang giỏ hàng.

2.4.14. Source code xóa sản phẩm giỏ hàng

❖ Chức năng : Xóa sản phẩm khỏi giỏ hàng của khách

```
package control;

import dao.DAO;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

@WebServlet(name="DeleteCartControl", urlPatterns={"/deletcart"})
public class DeleteCartControl extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String pid = request.getParameter("pid");
        DAO dao = new DAO();
        dao.deleteItem(pid);
        response.sendRedirect("cart");
    }

    HttpServlet methods. Click on the + sign on the left to edit the code.
}
```

- Thực hiện lấy id của sản phẩm muốn xóa qua getParameter và xóa sản phẩm bằng hàm dao.deleteItem và truyền id sản phẩm muốn xóa vào.

2.4.15. Source code thêm sản phẩm vào giỏ hàng

❖ Chức năng : Thêm sản phẩm vào giỏ hàng của khách

```

package control;

import dao.DAO;
import entity.Account;
import entity.Product;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;

@WebServlet(name="AddCartControl", urlPatterns={"/addcart"})
public class AddCartControl extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        HttpSession session = request.getSession();
        Account a = (Account) session.getAttribute("acc");
        int bid = a.getId();
        DAO dao = new DAO();
        String pid = request.getParameter("pid");
        Product i = dao.getProductByID(pid);
        int sellID = dao.getSellIDByProductID(pid);

        dao.insertItem(i.getName(), i.getImage(), Double.toString(i.getPrice()), i.getTitle(), i.getDescription(), "1", bid, sellID);
        response.sendRedirect("home");
    }
}

```

HttpServlet methods. Click on the + sign on the left to edit the code.

- Dùng session lấy Account hiện tại để sau đó lấy bid (id người mua), dùng getParamater để lấy pid (id của sản phẩm).
- Lấy sellID (id của người bán) bằng gestSellIDByProductID, sau đó thực hiện thêm vào giỏ hàng bằng hàm dao.insertItem và chuyển hướng về trang home.

2.4.16. Source code mua hàng

❖ Chức năng : Đặt đơn hàng cho khách

```

package control;
import dao.DAO;
import entity.*;
import entity.Product;
import java.io.*;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import java.util.*;
@WebServlet(name="BuyControl", urlPatterns={"/buy"})
public class BuyControl extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        HttpSession session = request.getSession();
        Account a = (Account) session.getAttribute("acc");
        int id = a.getId();
        DAO dao = new DAO();
        List<Product> list = dao.getProductByBuyID(id);
        String namecustomer = a.getUser();
        String phone = request.getParameter("phone");
        String address = request.getParameter("address");
        for(Product i: list){
            int sellID = dao.getSellIDByItemID(Integer.toString(i.getId()));
            dao.insertOrder(i.getName(), i.getImage(), Double.toString(i.getPrice()),
                i.getTitle(), i.getDescription(), "1", namecustomer, phone, address, sellID);
        }
        boolean check=true;
        request.setAttribute("check", check);
        for(Product i:list){dao.deleteItem(Integer.toString(i.getId()));}
        request.getRequestDispatcher("cart").forward(request, response);
    }
}

```

HttpServlet methods. Click on the + sign on the left to edit the code.

- Dùng session lấy Account hiện tại để sau đó lấy id người mua.
- Lấy list sản phẩm khách muốn mua thông qua dao.getProductByBuyID đồng thời lấy thêm số điện thoại và địa chỉ thông qua getParameter.
- Với mỗi sản phẩm khách đặt ta lấy id của người bán sau đó thêm vào cơ sở dữ liệu chứa đơn hàng của người bán.
- Biến check để xem khách đặt lần đầu hay chưa (Phục vụ cho việc hiển thị Order Success trên giao diện).
- Sau khi khách đặt thành công ta loại bỏ đơn của ra khỏi giỏ hàng bằng vòng for dao.deleteItem và chuyển hướng về giỏ hàng.

2.4.17. Source code class Account:

```

package entity;

public class Account {
    private int id;
    private String user;
    private String pass;
    private int isSell;
    private int isAdmin;

    public Account() {
    }

    public Account(int id, String user, String pass, int isSell, int isAdmin) {
        this.id = id;
        this.user = user;
        this.pass = pass;
        this.isSell = isSell;
        this.isAdmin = isAdmin;
    }

    public int getId() { ...3 lines }
    public void setId(int id) { ...3 lines }
    public String getUser() { ...3 lines }
    public void setUser(String user) { ...3 lines }
    public String getPass() { ...3 lines }
    public void setPass(String pass) { ...3 lines }
    public int getIsSell() { ...3 lines }
    public void setIsSell(int isSell) { ...3 lines }
    public int getIsAdmin() { ...3 lines }
    public void setIsAdmin(int isAdmin) { ...3 lines }
    @Override
    public String toString() { ...3 lines }
}

```

2.4.18. Source code class Order:

```

package entity;

public class Order {
    private int id;
    private String name;
    private String image;
    private double price;
    private String title;
    private String description;
    private String namecustomer;
    private String phone;
    private String address;

    public int getId() { ...3 lines }
    public Order(int id, String name, String image, double price, String title, String description,
        String namecustomer, String phone, String address) { ...11 lines }
    public void setId(int id) { ...3 lines }
    public String getName() { ...3 lines }
    public void setName(String name) { ...3 lines }
    public String getImage() { ...3 lines }
    public void setImage(String image) { ...3 lines }
    public double getPrice() { ...3 lines }
    public void setPrice(double price) { ...3 lines }
    public String getTitle() { ...3 lines }
    public void setTitle(String title) { ...3 lines }
    public String getDescription() { ...3 lines }
    public void setDescription(String description) { ...3 lines }
    public String getNamecustomer() { ...3 lines }
    public void setNamecustomer(String namecustomer) { ...3 lines }
    public String getPhone() { ...3 lines }
    public void setPhone(String phone) { ...3 lines }
    public String getAddress() { ...3 lines }
    public void setAddress(String address) { ...3 lines }
    @Override
    public String toString() { ...3 lines }
}

```

2.4.19. Source code class Product:

```
package entity;

import dao.DAO;

public class Product{
    private int id;
    private String name;
    private String image;
    private double price;
    private String title;
    private String description;

    public Product() {...2 lines }
    public Product(int id, String name, String image, double price, String title, String description) {...8 lines }
    public int getId() {...3 lines }
    public void setId(int id) {...3 lines }
    public String getName() {...3 lines }
    public void setName(String name) {...3 lines }
    public String getImage() {...3 lines }
    public void setImage(String image) {...3 lines }
    public double getPrice() {...3 lines }
    public void setPrice(double price) {...3 lines }
    public String getTitle() {...3 lines }
    public void setTitle(String title) {...3 lines }
    public String getDescription() {...3 lines }
    public void setDescription(String description) {...3 lines }
    @Override
    public String toString() {...3 lines }
}
```

2.5 THIẾT KẾ GIAO DIỆN HỆ THỐNG

Phần giao diện phải hoạt động tốt, hiệu quả, không trục trặc về thông tin, hạn chế sai sót trong các thao tác kỹ thuật.

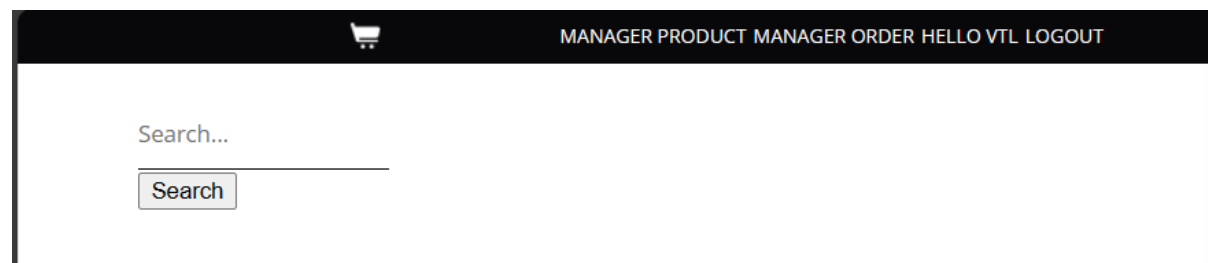
2.5.1. Form Header



Search...

Search

- Manager Product, Manager Order, giỏ hàng sẽ hiện ra đối với người bán sản phẩm.
- Người đăng nhập bình thường sẽ có giỏ hàng.

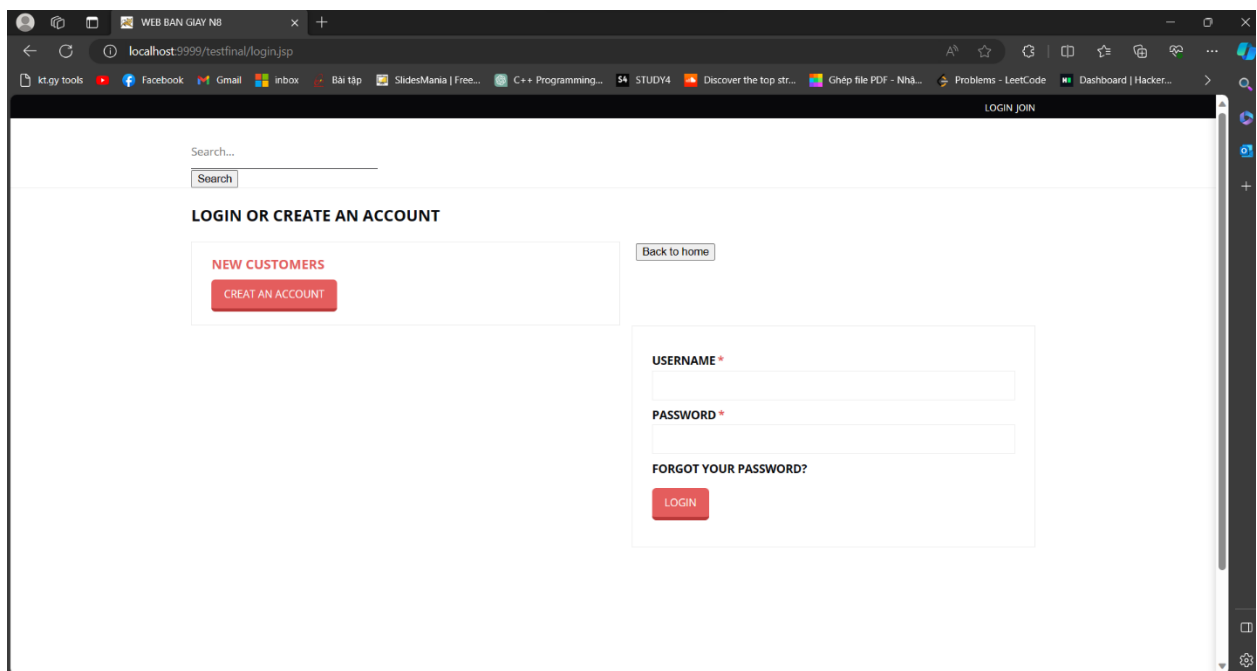




Chức năng nút lệnh:

Tên nút lệnh	Ý nghĩa	Sự kiện
Login	Vào trang đăng nhập	Click
Join	Vào trang đăng kí	Click
Search	Tìm kiếm sản phẩm dựa vào phần đã nhập bên trên	Click
Icon giỏ hàng	Vào giỏ hàng	Click
Manager Order	Vào trang quản lí đơn hàng	Click
Manager Product	Vào trang quản lí những sản phẩm của đối tượng bán	Click

2.5.2. Form đăng nhập



Chức năng nút lệnh:

Tên nút lệnh	Ý nghĩa	Sự kiện
LOGIN	Đăng nhập dựa trên USERNAME và PASSWORD	Click
CREAT AN ACCOUNT	Vào trang đăng kí	Click

Back to home	Trở về trang chủ	Click
--------------	------------------	-------

2.5.3. Form Đăng kí

Chức năng nút lệnh:

Tên nút lệnh	Ý nghĩa	Sự kiện
Sign up	Đăng kí tài khoản dự trên user và pass cung cấp	Click
Back to home	Trở về trang chủ	Click

Form trang home

Chức năng nút lệnh:

Tên nút lệnh	Ý nghĩa	Sự kiện
DETAILS	Vào trang detail của sản phẩm tương ứng.	Click

2.5.4. Form Chi tiết sản phẩm

Search...

Search

[HOME](#)

AIR JORDAN 1 LOW



Air Jordan 1 Low - sự kết hợp hoàn hảo giữa thiết kế thể thao và phong cách đường phố. Chất liệu da cấp cao và đế ngoài độc đáo mang lại cảm giác thoải mái và ổn định. Với logo Jumpman nổi tiếng, đây không chỉ là đôi giày, mà là biểu tượng của phong cách cá nhân. Sẵn sàng đồng hành cùng bạn từ phòng tập đến đường phố, Air Jordan 1 Low là sự lựa chọn hoàn hảo cho ngày hôm nay.

100.0 \$



Air Jordan 1 Low - sự kết hợp hoàn hảo giữa thiết kế thể thao và phong cách đường phố. Chất liệu da cao cấp và đế ngoài độc đáo mang lại cảm giác thoải mái và ổn định. Với logo Jumpman nổi tiếng, đây không chỉ là đôi giày, mà là biểu tượng của phong cách cá nhân. Sẵn sàng đồng hành cùng bạn từ phòng tập đến đường phố, Air Jordan 1 Low là sự lựa chọn hoàn hảo cho ngày hôm nay.

100.0 \$

ADD TO CART

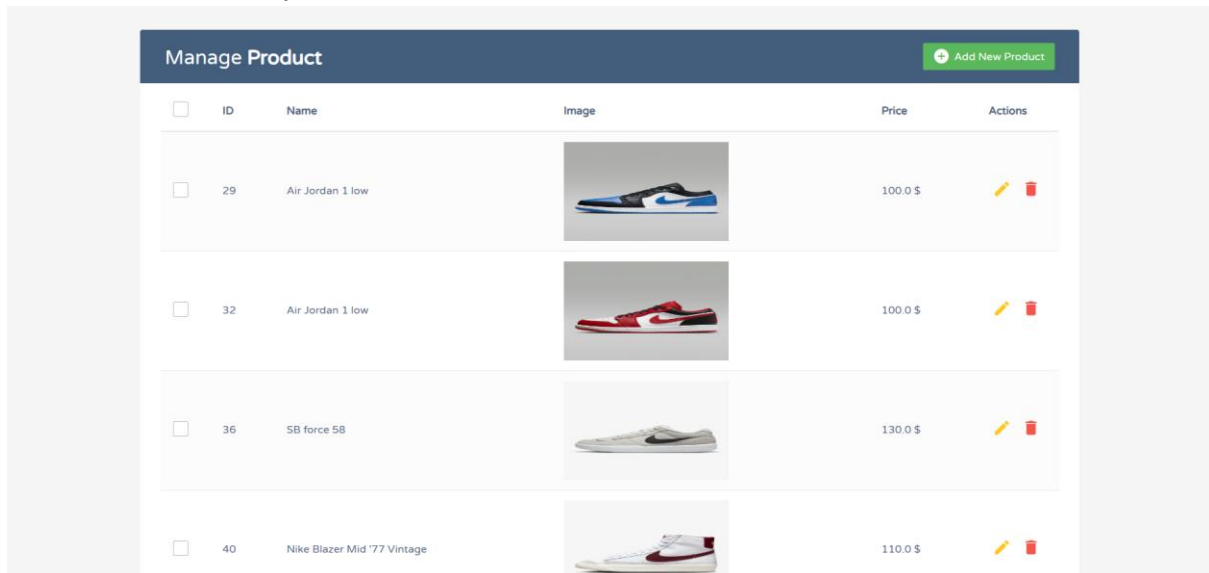
[BACK TO HOME](#)



Chức năng nút lệnh:

Tên nút lệnh	Ý nghĩa	Sự kiện
Back to home	Về trang chủ	Click
Add to cart	Thêm sản phẩm vào giỏ hàng (khi đã đăng nhập)	
Login to cart	Đến trang đăng nhập để đăng nhập để mua hàng	

2.5.5. Form Quản Lý Sản Phẩm



Chức năng nút lệnh:

Tên nút lệnh	Ý nghĩa	Sự kiện
Add New Product	Thêm sản phẩm	Click
Icon bút	Chỉnh sửa thông tin sản phẩm	Click
Icon thùng rác	Xóa sản phẩm	Click
Back to home	Trở về trang chủ	Click

2.5.6. Form Trang Edit

Add Product

ID
29

Name
Air Jordan 1 low

Image
https://static.nike.com/a/images/t_PDP_1728_v1/f_auto,q_auto:eco,u_126ab3f

Price
100.0

Title
Air Jordan 1 low




Description
Air Jordan 1 Low - sự kết hợp hoàn hảo giữa thiết kế thể thao và phong cách đường phố. Chất liệu da cao cấp và đế ngoài độc đáo mang lại cảm giác thoải

Edit

Chức năng nút lệnh:

Tên nút lệnh	Ý nghĩa	Sự kiện
Edit	Sửa thông tin sản phẩm	Click
Back to home	Trở về trang chủ	Click

2.5.7. Form Quản lý giỏ hàng

SẢN PHẨM	ĐƠN GIÁ	XÓA
 Air Jordan 1 low	100.0	Delete
 Air Jordan 1 low	100.0	Delete
 SB force 58	130.0	Delete

Tổng tiền hàng:330




Phone...

Address...

Mua Hàng

Home

ORDER SUCCESS!!!

SẢN PHẨM	ĐƠN GIÁ	XÓA
 Air Jordan 1 low	100.0	Delete
 Air Jordan 1 low	100.0	Delete
 SB force 58	130.0	Delete

Tổng tiền hàng:0

Phone...

Address...




Mua Hàng

Home

Chức năng nút lệnh:

Tên nút lệnh và ô input	Ý nghĩa	Sự kiện
Delete	Xóa sản phẩm khỏi giỏ	Click
Mua hàng	Thực hiện đặt hàng	Click
Phone	Ô chứa text	Nhập
Address	Ô chứa text	Nhập
Home	Trở về trang chủ	Click

2.5.8. Form Quản lý đơn hàng

NAME CUSTOMER	PHONE	ADRESS	PRODUCT	PRICE	DELETE
vt1	0346842376	Vinh Tho- Hương Mạc- Từ Sơn- Bắc Ninh	 SB force 58	130.0	Delete
vt1	0346842376	Vinh Tho- Hương Mạc- Từ Sơn- Bắc Ninh	 Air Jordan 1 low	100.0	Delete
vt1	0346842376	Vinh Tho- Hương Mạc- Từ Sơn- Bắc Ninh	 Air Jordan 1 low	100.0	Delete

Home

Chức năng nút lệnh:

Tên nút lệnh và ô input	Ý nghĩa	Sự kiện
Delete	Xóa đơn đặt	Click
Home	Trở về trang chủ	Click

3. KẾT LUẬN

3.1. KẾT LUẬN ĐỀ TÀI

➤ Ưu điểm:

- Rút ngắn được thời gian mua hàng, quản lý sản phẩm khi sử dụng website.
- Sử dụng website vào các công việc tìm kiếm các thông tin chi tiết về sản phẩm, giá cả, mẫu mã theo sở thích cá nhân và mua hàng một cách nhanh chóng và thuận tiện. Việc lưu trữ sẽ đơn giản, không cần có không gian lưu trữ lớn, các thông tin bán hàng hay quản lý nhân viên sẽ dễ dàng và thuận tiện để sử dụng hơn.
- Việc thống kê hoá đơn, các sản phẩm sẽ dễ dàng và thuận tiện hơn.
- Với chức năng xử lý hệ thống mới sẽ giúp cải thiện hiệu suất công việc của người bán hàng. Đồng thời giúp giảm số lượng nhân viên quản lý, tránh tình trạng dư thừa.

➤ Nhược điểm:

- Do kinh nghiệm và trình độ còn nhiều hạn chế nên bài toán “Xây dựng Website bán giày” còn tồn tại nhiều thiếu sót.
- Giao diện website còn đơn giản.
- Cần bổ sung thêm nhiều tiện ích người dùng hơn nữa.

3.2. LỜI KẾT

Xây dựng website bán hàng nói chung và website bán giày nói riêng không chỉ là việc xây dựng đơn thuần, mà nó đòi hỏi một cách có hệ thống các giai đoạn khảo sát, phân tích thiết kế phải được tiến hành trước đó. Từ đó đem đến cho người dùng một website tiện ích, cơ bản đáp ứng được nhu cầu tìm kiếm và mua sắm online.

Đứng trước xu thế phát triển của công nghệ thông tin như vũ bão, cũng như đáp ứng nhu cầu mua sắm online ngày càng tăng hiện nay thì xây dựng các website bán hàng là điều không thể thiếu, đây là một đề tài mang tính thực tế cao. Qua việc nghiên cứu xây dựng đề tài này, ngoài việc giúp chúng em củng cố cho em kiến thức về lập trình thì nó cũng cung cấp cho em thêm là làm thế nào có thể xây dựng được một website hoàn chỉnh. Thông qua việc xây dựng website này đã giúp em đã có thể tự tin xây dựng được các website tương tự như: website bán quần áo, website bán đồ nội thất, ...

Tài liệu tham khảo

1. Các video bài giảng trên Youtube.
2. Các template tham khảo trên Google.
3. Slide bài giảng “Lập trình hướng đối tượng” của GV Nguyễn Mạnh Sơn - GV Học viện Công nghệ Bưu chính viễn thông,