

ĐẠI HỌC UEH

ĐẠI HỌC KINH TẾ THÀNH PHỐ HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN KINH DOANH



BÁO CÁO DỰ ÁN NGHIÊN CỨU

Đề tài

XÂY DỰNG WEBSITE TMĐT CHO CỬA HÀNG KINH
DOANH VÒNG Ý - USBIBRACELET

BỘ MÔN: PHÁT TRIỂN ỨNG DỤNG TMĐT

Giảng viên: Nguyễn Thành Huy

Mã lớp học phần: 24C1INF50902701

Nhóm sinh viên thực hiện

Trần Thu Huyền	31221023150
Đoàn Thị Linh Chi	31221026387
Nguyễn Đình Minh	31221020867
Phù Lữ Phúc	31221023196
Phạm Thị Mỹ Tâm	31221026441
Lê Nguyễn Thúy Vy	31221024274

Thành phố Hồ Chí Minh, ngày 14 tháng 12 năm 2024

LỜI CẢM ƠN

Lời đầu tiên xin phép chúng em – Nhóm 1 gửi lời cảm ơn tới Đại học Kinh tế TP.Hồ Chí Minh đã đưa bộ môn “Phát triển ứng dụng Thương mại điện tử” vào chương trình học. Đặc biệt cảm ơn sâu sắc tới Thầy – TS.Nguyễn Thành Huy đã giảng dạy và truyền đạt những kiến thức quý giá hỗ trợ nhóm chúng em trong suốt quá trình học tập cũng như hoàn thành bài đồ án **“XÂY DỰNG WEBSITE BÁN HÀNG VÒNG TAY Ý USBIBRACELET”**.

Thông qua sự tận tình trong những bài giảng và ứng dụng thực tế của Thầy, chúng em đã học hỏi được nhiều kiến thức về môn Phát triển ứng dụng Thương mại điện tử, hiểu sâu hơn về chuyên ngành mình đang theo học từ cơ bản đến nâng cao, hiểu được cách vận hành một trang web, ngôn ngữ lập trình, ứng dụng hỗ trợ xây dựng nên trang web, chỉnh sửa trang web sao cho thân thiện với người dùng,...

Sau cùng, vì còn thiếu kinh nghiệm nên bài đồ án không tránh khỏi những thiếu sót, nhóm chúng em rất mong sẽ nhận được sự góp ý của Thầy để hoàn thiện một cách tốt nhất. Với những gì Thầy chia sẻ giúp nhóm có thể phát triển hơn trong lĩnh vực mà mình theo đuổi trong tương lai. Một lần nữa, nhóm chúng em xin chân thành cảm ơn Thầy Nguyễn Thành Huy đã tận tình hướng dẫn hoàn thành bài đồ án này và chúc Thầy luôn nhiệt huyết với sự nghiệp cao cả, gặt hái được nhiều thành công.

Nhóm 1

LỜI CAM ĐOAN

Nhóm chúng em xin cam đoan đây là công trình nghiên cứu của riêng nhóm. Các số liệu, tài liệu, kết quả được nêu trong đồ án là trung thực, có nguồn gốc và xuất xứ rõ ràng, không trùng lắp hay sao chép bất cứ công trình khoa học nào đã công bố. Việc tham khảo các nguồn tài liệu đã được trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

Nhóm thực hiện

Nhóm 1

MỤC LỤC

LỜI CẢM ƠN.....	2
LỜI CAM ĐOAN.....	3
PHÂN CÔNG CÁC THÀNH VIÊN.....	7
CHƯƠNG A: GIAO DIỆN NGƯỜI DÙNG.....	9
 1. Đăng nhập.....	9
1.1. Giao diện.....	9
1.1.2. Các tương tác.....	9
1.2. Chức năng.....	10
 2. Layout (Header, footer, sidebar).....	12
2.1. Header.....	12
2.2 Footer.....	14
 3. Trang chủ.....	14
3.1. Giao diện.....	14
3.2. Chức năng.....	18
 4. Cửa hàng.....	20
4.1. Giao diện.....	20
4.2. Chức năng.....	24
 5. Chi tiết sản phẩm.....	27
5.1. Giao diện.....	27
5.2. Chức năng.....	33
 6. Giỏ hàng.....	36
6.1. Giao diện.....	36
6.2. Chức năng.....	38
 7. Thanh toán.....	44
7.1. Giao diện.....	44
7.1.1 Trang thanh toán mục vận chuyển.....	44
7.1.2 Trang thanh toán mục giao hàng.....	52
7.1.3 Trang thanh toán mục thanh toán.....	55
7.1.4 Giao diện trang “Xác nhận đơn hàng thành công”.....	58
7.2. Chức năng.....	58
7.2.1 Điện thoại mua hàng.....	58
7.2.2 Chọn phương thức vận chuyển.....	66
7.2.3 Chọn phương thức thanh toán.....	69
7.2.4 Thông báo thanh toán thành công.....	71
 8. Lọc sản phẩm.....	72
8.1 Giao diện.....	72
8.2 Chức năng.....	73

9. Tìm kiếm sản phẩm.....	79
9.1 Giao diện.....	79
9.2 Chức năng.....	79
10. Tin tức.....	84
10.1 Giao diện.....	84
10.1.1 Danh sách tin tức.....	84
10.2. Chức năng.....	89
10.2.1 Xem tổng quát trang Blog.....	89
10.2.2 Xem chi tiết từng blog.....	90
11. Về chúng tôi.....	91
12. Liên hệ.....	92
12.1 Các thành phần từ trái sang.....	92
12.2 Các tương tác.....	92
CHƯƠNG B: GIAO DIỆN QUẢN LÝ.....	94
1. Quản lý người dùng.....	94
1.1 Giao diện.....	94
1.1.1 Các thành phần từ trái sang phải, từ trên xuống dưới:.....	94
1.1.2 Các tương tác.....	95
1.2 Chức năng.....	97
1.2.1 User Management.....	97
1.2.2 User management detail.....	99
2. Quản lý sản phẩm.....	103
2.1 Giao diện.....	103
2.1.1 Các thành phần từ trái sang phải, từ trên xuống dưới:.....	103
2.1.2 Các tương tác.....	104
2.2 Chức năng.....	107
3. Quản lý tin tức.....	112
3.1 Giao diện.....	112
3.1.1 Các thành phần từ trái sang phải, từ trên xuống dưới:.....	112
3.1.2 Các tương tác.....	113
3.2 Chức năng.....	116
4. Quản lý đơn hàng.....	119
4.1 Giao diện.....	119
4.2 Chức năng.....	121
CHƯƠNG C: KẾT NỐI DATABASE.....	129
1. Tạo cơ sở dữ liệu.....	129
2. Tạo các bảng.....	129
3. Thêm dữ liệu vào bảng.....	131

4. Tạo các mối quan hệ giữa các bảng bằng khóa ngoại.....	132
5. Kết nối database với Xampp.....	135
6. ERD.....	136
7. Mô tả mối quan hệ.....	136
TÀI LIỆU THAM KHẢO.....	139
PHỤ LỤC.....	140

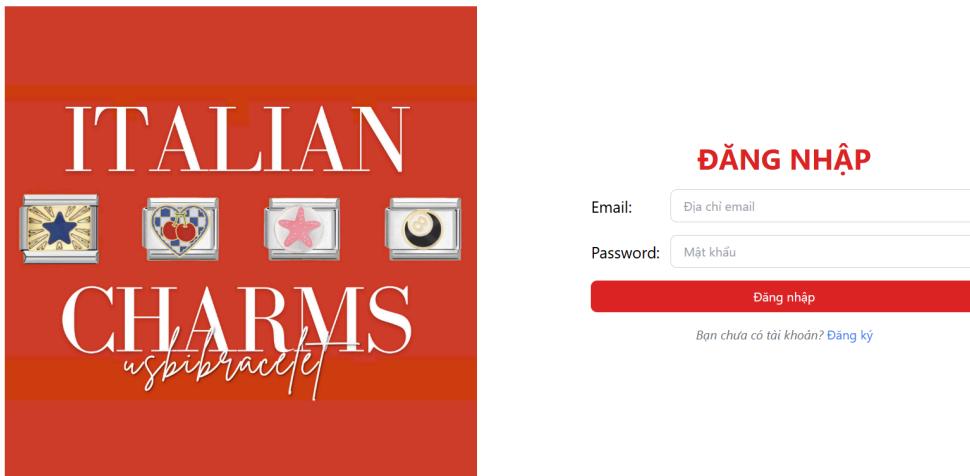
PHÂN CÔNG CÁC THÀNH VIÊN

STT	Họ và tên	Công việc	Hoàn thành
1	Đoàn Thị Linh Chi 31221026387 (Nhóm trưởng)	<ul style="list-style-type: none"> - Figma: Font chữ, Giỏ hàng, Thanh toán. - Front-end: Danh sách sản phẩm, Quản lý đơn hàng. - Back-end: Đăng nhập, Trang chủ, Lọc, Tìm kiếm, Chi tiết sản phẩm, Giỏ hàng, Quản lý sản phẩm, Quản lý đơn hàng. 	100%
2	Trần Thu Huyền 31221023150	<ul style="list-style-type: none"> - Figma: Trang chủ, Liên hệ. - Front-end: Đăng nhập, Header, Footer, Trang chủ, Danh sách sản phẩm, Tin tức, Về chúng tôi, Liên hệ, Giao diện admin, Quản lý người dùng, Quản lý sản phẩm, Quản lý tin tức, Quản lý đơn hàng. - Chuẩn bị bài viết. 	100%
3	Nguyễn Đình Minh 31221020867	<ul style="list-style-type: none"> - Figma: Đăng nhập/Đăng ký. - Database: Thiết kế, kết nối, vẽ ERD, nhập dữ liệu. - Front-end: Đăng nhập, Thanh toán thành công - Back-end: Quản lý sản phẩm. 	100%

4	Lê Nguyễn Thúy Vy 31221024274	<ul style="list-style-type: none"> - Figma: Cửa hàng. - Front-end: Danh sách sản phẩm, Thanh toán, Quản lý đơn hàng, Tìm kiếm. - Thiết kế banner, format. 	100%
5	Phù Lữ Phúc 31221023196	<ul style="list-style-type: none"> - Figma: Layout, Tài khoản - Front-end: Layout, Quản lý tin tức. - Back-end: Danh sách sản phẩm (cửa hàng), Tin tức, Thanh toán, Quản lý người dùng, Quản lý tin tức. 	100%
6	Phạm Thị Mỹ Tâm 31221026441	<ul style="list-style-type: none"> - Figma: Chi tiết sản phẩm, Tin tức, Chi tiết tin tức. - Front-end: Đăng nhập, Chi tiết sản phẩm, Giỏ hàng, Lọc, Tin tức, Chi tiết tin tức, Quản lý người dùng, Quản lý tin tức, Quản lý sản phẩm. - Chuẩn bị sản phẩm. 	100%

CHƯƠNG A: GIAO DIỆN NGƯỜI DÙNG

1. Đăng nhập



Hình. Trang đăng nhập

1.1. Giao diện

Các thành phần từ trái sang, từ trên xuống dưới

- Ảnh charm của Usbibracelet
- Phần nhập địa chỉ email
- Phần nhập mật khẩu
- Nút Đăng nhập
- Mục Bạn chưa có tài khoản? Đăng ký

1.1.2. Các tương tác

Nút Đăng nhập

- Mô tả: Sau khi người dùng nhập tài khoản và mật khẩu, hệ thống sẽ đối chiếu thông tin với Database, nếu tài khoản hợp lệ sẽ chuyển người dùng đến trang cá nhân tương ứng, ngược lại sẽ báo lỗi “Sai tài khoản hoặc mật khẩu”.
- Hiệu ứng: Đổi màu khi di chuột vào.
- Sự kiện: Chuyển người dùng đến trang cá nhân tương ứng nếu tài khoản và mật khẩu hợp lệ.

Nút Đăng ký

- Mô tả: là đường dẫn (link) tới trang đăng ký

- Hiệu ứng: Xuất hiện một đoạn thăng gạch dưới tên mục đăng nhập khi người dùng rê chuột đến.
- Sự kiện: Khi người dùng nhấn chọn, lập tức sẽ được điều phổi đến trang đăng nhập.

1.2. Chức năng

Bước 1: Người dùng truy cập **login.php** và nhập email/mật khẩu vào form đăng nhập.

```
<form class="w-4/5" action="include/login.inc.php" method="post">
  <div class="mb-4 grid grid-cols-5 items-center gap-3">
    <label htmlFor="email" class="text-xl"> Email: </label>
    <input
      id="email"
      name="email"
      class="col-span-4 rounded-lg border border-gray-300 px-4 py-2
      focus:outline-none focus:ring-2 focus:ring-red-600"
      placeholder="Địa chỉ email"
      type="email"
    />
  </div>
  <div class="mb-4 grid grid-cols-5 items-center gap-3">
    <label htmlFor="password" class="text-xl"> Password: </label>
    <input
      id="password"
      name="password"
      class="col-span-4 rounded-lg border border-gray-300 px-4 py-2
      focus:outline-none focus:ring-2 focus:ring-red-600"
      placeholder="Mật khẩu"
      type="password"
    />
  </div>
  <button
    class="w-full rounded-lg bg-red-600 py-2 text-white hover:bg-red-700"
    type="submit"
  >
```

```
Đăng nhập  
</button>  
</form>
```

- **action="include/login.inc.php"**: Thuộc tính action trong form này chỉ định rằng khi người dùng nhấn nút "Đăng nhập", form sẽ gửi dữ liệu đến file login.inc.php để xử lý.
- **method="post"**: Form gửi dữ liệu qua phương thức POST, giúp bảo mật thông tin (như mật khẩu).

Khi nhấn nút "Đăng nhập", form gửi yêu cầu POST đến **login.inc.php**.

Bước 2: login.inc.php kiểm tra nếu phương thức là POST và lấy giá trị email/mật khẩu từ form.

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $email = $_POST["email"];  
    $password = $_POST["password"];
```

Bước 3: Kết nối đến cơ sở dữ liệu và thực thi câu truy vấn để kiểm tra tài khoản có tồn tại không.

```
require_once "db.inc.php";  
$query = "SELECT * FROM account WHERE email = :email;";  
$stmt = $pdo->prepare($query);  
$stmt->bindParam(":email", $email);  
$stmt->execute();
```

Nếu tài khoản tồn tại, so sánh mật khẩu người dùng nhập với mật khẩu trong cơ sở dữ liệu.

Nếu mật khẩu đúng, lưu thông tin người dùng vào session và chuyển hướng:

- **Nếu là admin**, chuyển hướng đến trang quản lý (user-management.php).
- **Nếu là người dùng thường**, chuyển hướng đến trang chủ (index.php).

```
if ($result) {  
    if ($password == $result["password"]) {  
        $_SESSION["user_id"] = $result["id"];
```

```

$_SESSION["user_name"] = $result["firstname"] . " " . $result["lastname"];

if ($result['role'] == 'admin') {

    header("Location: ../user-management.php");

} else {

    header("Location: ../index.php");

}

exit();

} else {

    die("Invalid password.");

}

} else {

    die("Account not found.");

}

}

```

Nếu có lỗi (mật khẩu sai hoặc tài khoản không tồn tại), hiển thị thông báo lỗi.

```

catch (Exception $e) {

    die("Query failed: " . $e->getMessage());

}

```

Nếu không phải **POST**, chuyển hướng về trang chủ ([index.php](#)).

```

else {

    header("Location: ../index.php");

    die();
}

```

2. Layout (Header, footer, sidebar)

2.1. Header

- Trước khi đăng nhập:



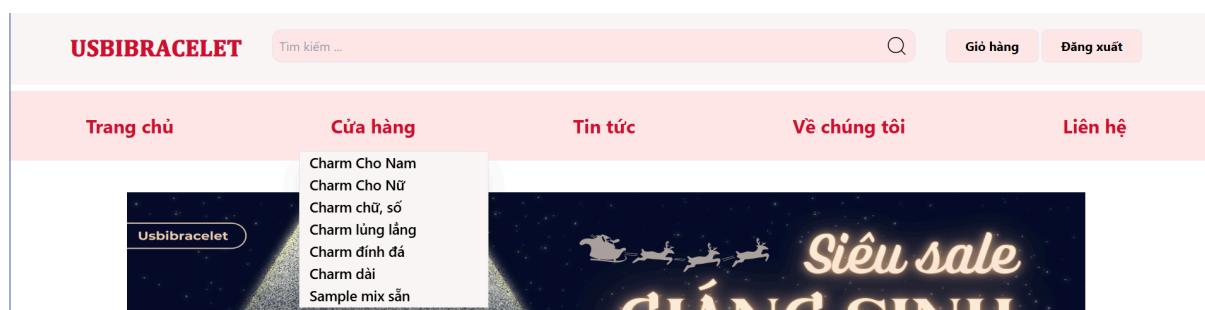
Hình - Header trước khi đăng nhập

Các thành phần trong phần Top Header:

1. Tên cửa hàng Usbibracelet
2. Search bar
3. 2 nút Đăng nhập và Đăng ký

Các thành phần trong thanh Navigation từ trái sang:

1. Mục trang chủ
2. Mục Cửa hàng sẽ có dropdown để phân loại Category
3. Mục Tin tức
4. Mục Về chúng tôi
5. Mục Liên hệ
- Sau khi đăng nhập:



Hình - Header sau khi đăng nhập

Các thành phần trong phần Top Header:

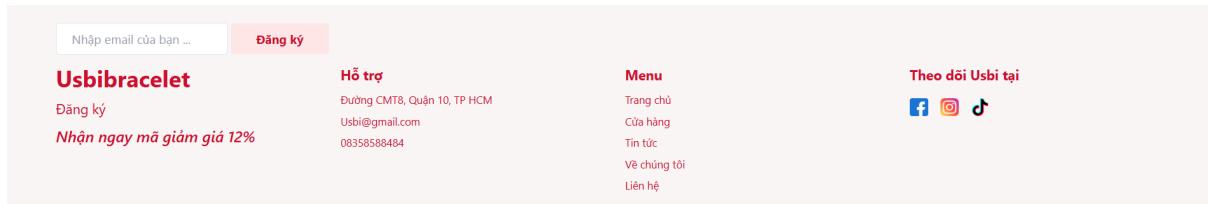
1. Tên cửa hàng Usbibracelet
2. Search bar
3. 2 nút Giỏ hàng và Đăng ký

Các thành phần trong thanh Navigation từ trái sang:

6. Mục trang chủ
7. Mục Cửa hàng sẽ có dropdown để phân loại Category

8. Mục Tin tức
9. Mục Về chúng tôi
10. Mục Liên hệ

2.2 Footer



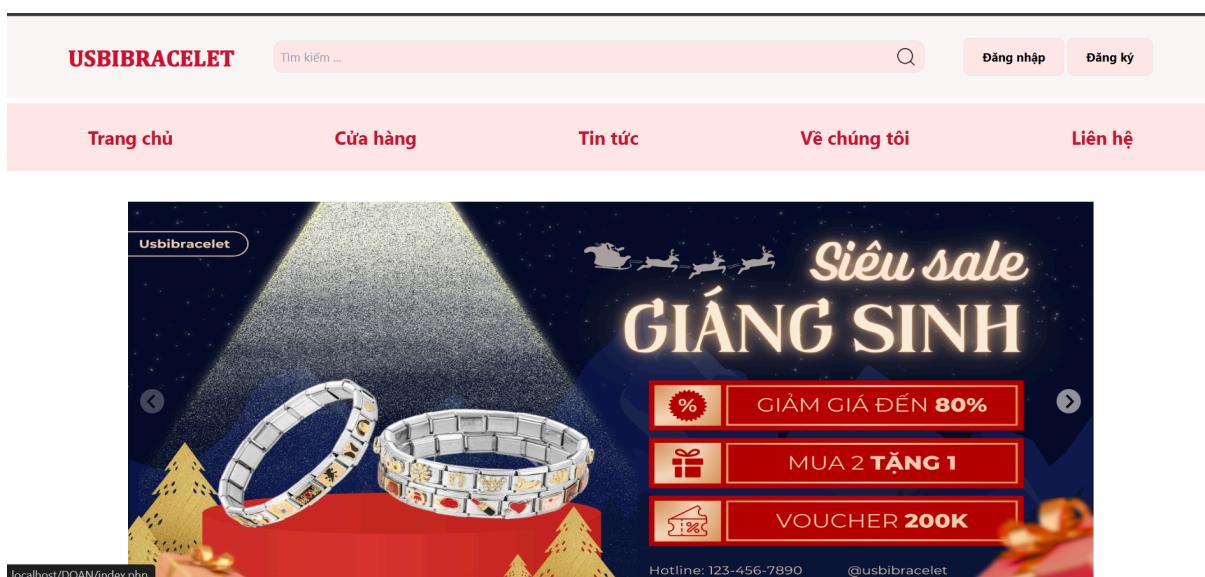
Hình - Footer

Các thành phần trong phần Footer được chia thành 4 phần từ trái sang phải:

1. Ở input để nhập email nhận thông báo và Những thông tin của Usbibracelet:
Nhận ngay mã giảm giá 12%
2. Phần Hỗ trợ: Bao gồm thông tin hỗ trợ như email , vị trí và số điện thoại của cửa hàng
3. Phần Menu điều hướng đến các trang
4. Phần Theo dõi Usbi tại : Instagram, Facebook, Tiktok.

3. Trang chủ

3.1. Giao diện



Hình. Trang chủ cửa hàng (1)

Phân loại

DANH MỤC SẢN PHẨM



Charm cho nữ



Charm cho nam



Charm chữ, số



Charm lủng lẳng



Charm đính đá



Charm dài



Sample mix sǎn

Phân loại

DANH MỤC SẢN PHẨM



Charm cho nữ



Charm cho nam



Charm chữ, số



Charm lùng lâng



Charm đính đá



Charm dài



Sample mix sẫn

Hình. Trang chủ cửa hàng (2)

Sản phẩm

BEST SELLER



Logo Ferrari

59.000 VND

★★★★★ (25)



Hoa Tulip hồng

55.000 VND

★★★★★ (30)



Vòng tay Sample Family

499.000 VND

★★★★★ (29)



I Love My Family

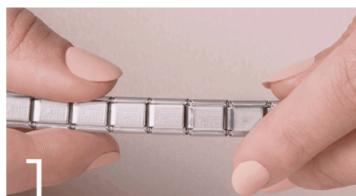
89.000 VND

★★★★★ (17)

Hình. Trang chủ cửa hàng (3)

Hướng dẫn

CÁCH SỬ DỤNG



Mở rộng mối nối của vòng đeo tay



Nắm bắt hai đầu của charm muốn tách và tháo ra



Chèn và nối các Charms mới lại với nhau

Hình. Trang chủ cửa hàng (4)



Hình. Trang chủ cửa hàng (5)



THE LOVE OF MINE

Tặng đồ trang sức tôn vinh gia đình không chỉ là một món quà có giá trị mà còn là một hành động mang ý nghĩa sâu sắc, giàu cảm xúc và đóng đầy những giá trị tinh thần: nó không dừng lại ở khía cạnh vật chất, mà còn vượt lên để trở thành biểu tượng hữu hình của tình yêu thương, sự gắn bó keo sơn và lòng biết ơn chân thành đối với những người thân yêu trong gia đình.

[ĐỌC NGAY](#)

Hình. Trang chủ cửa hàng (6)

KHÁM PHÁ USBIBRACELET



LOVE X USBI
[Khám phá ngay](#)



BRACESUCCESS
[Khám phá ngay](#)



MY MUSE
[Khám phá ngay](#)

Hình. Trang chủ cửa hàng (7)



Miễn phí vận chuyển

Miễn phí giao hàng trên toàn quốc với
mọi giá trị đơn hàng



Hỗ trợ 24/7

Đội ngũ chuyên nghiệp luôn hỗ trợ và
tư vấn hoàn toàn miễn phí



Bảo hành tuyệt đối

Bảo hành sản phẩm uy tín với thời
gian tối đa lên đến 1 năm

Đăng ký

Usbibracelet

Đăng ký

Hỗ trợ

Menu

[Nhận ngay mã giảm giá 12%](#)

[Đường CMT8, Quận 10, TP HCM](#)
[Usbi@gmail.com](#)
[08358588484](#)

[Trang chủ](#)
[Cửa hàng](#)
[Tin tức](#)
[Về chúng tôi](#)
[Liên hệ](#)

Theo dõi Usbi tại

f i d

Hình. Trang chủ cửa hàng (8)

Giao diện trang chủ được thể hiện từ trên xuống dưới bao gồm:

- Banner cửa hàng, sự kiện
- Danh mục sản phẩm
- Sản phẩm Best-sellers
- Hướng dẫn đeo vòng tay
- Hình ảnh cửa hàng
- Tin tức
- Cam kết cửa hàng

3.2. Chức năng

- Khởi tạo session và kiểm tra trạng thái người dùng

```
<?php
session_start();

// Kiểm tra xem người dùng đã đăng nhập chưa
if (isset($_SESSION["user_name"])) {
    // Người dùng đã đăng nhập
    $user_name = $_SESSION["user_name"];
} else {
    // Người dùng chưa đăng nhập
    $user_name = null;
}
?>
```

- Hiển thị các nút và các liên kết theo trạng thái đăng nhập

```
<?php
if (isset($_SESSION["user_name"])) {
    // Nếu người dùng đã đăng nhập
    echo "<a href='cart.php'><button class='btn'>Giỏ hàng</button></a>";
    echo "<form method='post' action='include/logout.inc.php'>
        <button class='btn'>Đăng xuất</button>
    </form>";
} else {
    // Nếu người dùng chưa đăng nhập
    echo "<button onclick=\"window.location.href='login.php'" class='btn'>Đăng
nhập</button>";
    echo "<button onclick=\"window.location.href='register.php'" class='btn'>Đăng
ký</button>";
}
?>
```

- Hiển thị các danh mục sản phẩm

```
$categories = [
    ['id' => 1, 'name' => 'Charm cho nữ'],
    ['id' => 2, 'name' => 'Charm cho nam'],
    ['id' => 3, 'name' => 'Charm chữ, số'],
    // Thêm các danh mục khác
];

foreach ($categories as $category) {
    echo "<div><a href='product-list.php?categoryId={$category['id']}>{$category['name']}

```

- Chức năng đăng xuất

```

<?php
// Khi người dùng nhấn "Đăng xuất"
if (isset($_POST['logout'])) {
    session_unset(); // Xóa tất cả các session
    session_destroy(); // Hủy session
    header("Location: index.php"); // Chuyển hướng về trang chủ
    exit();
}
?>

```

4. Cửa hàng

4.1. Giao diện



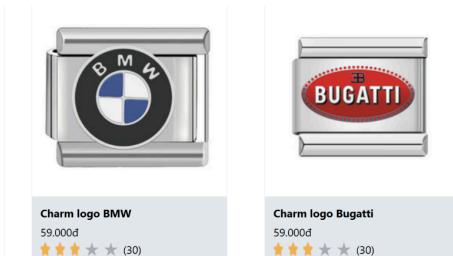
Hình - Giao diện trang sản phẩm (1)

Sắp xếp theo

TẤT CẢ SẢN PHẨM

Charm hoa Tulip trắng 55.000đ ★★★★☆ (30)	Charm hoa Tulip tím 55.000đ ★★★★☆ (30)	Charm hoa Tulip hồng 55.000đ ★★★★☆ (30)	Charm hoa Tulip đỏ 55.000đ ★★★★☆ (30)
Charm hoa Sen 55.000đ ★★★★☆ (30)	Charm hoa Mini 55.000đ ★★★★☆ (30)	Charm hoa Ly 55.000đ ★★★★☆ (30)	Charm hoa Hướng Dương 55.000đ ★★★★☆ (30)

Hình - Giao diện trang sản phẩm (2)



CÁCH SỬ DỤNG



Hình - Giao diện trang sản phẩm (3)

Các chi tiết chính của trang giao diện “Cửa hàng” gồm:

Danh mục sản phẩm

Các danh mục sản phẩm được hiển thị với hình Charm đại diện và tên tương ứng giúp khách hàng có cái nhìn chi tiết hơn về các loại sản phẩm, dễ dàng tìm kiếm và giúp tối ưu trải nghiệm người dùng.

- Charm cho nữ
- Charm cho nam
- Charm chữ, số
- Charm lủng lăng
- Charm đính đá
- Charm dài
- Sample mix sǎn

DANH MỤC SẢN PHẨM



Charm cho nữ



Charm cho nam



Charm chữ, số



Charm lủng lăng



Charm đính đá



Charm dài



Sample mix sǎn

Banner khuyến mãi cuối năm

Banner được đặt giữa trang nhằm thông báo đến khách hàng những chương trình đặc biệt của Usbibracelet

Nút CTA "XEM THÊM": Nút màu đỏ nổi bật, khuyến khích người dùng tìm hiểu thêm chi tiết.



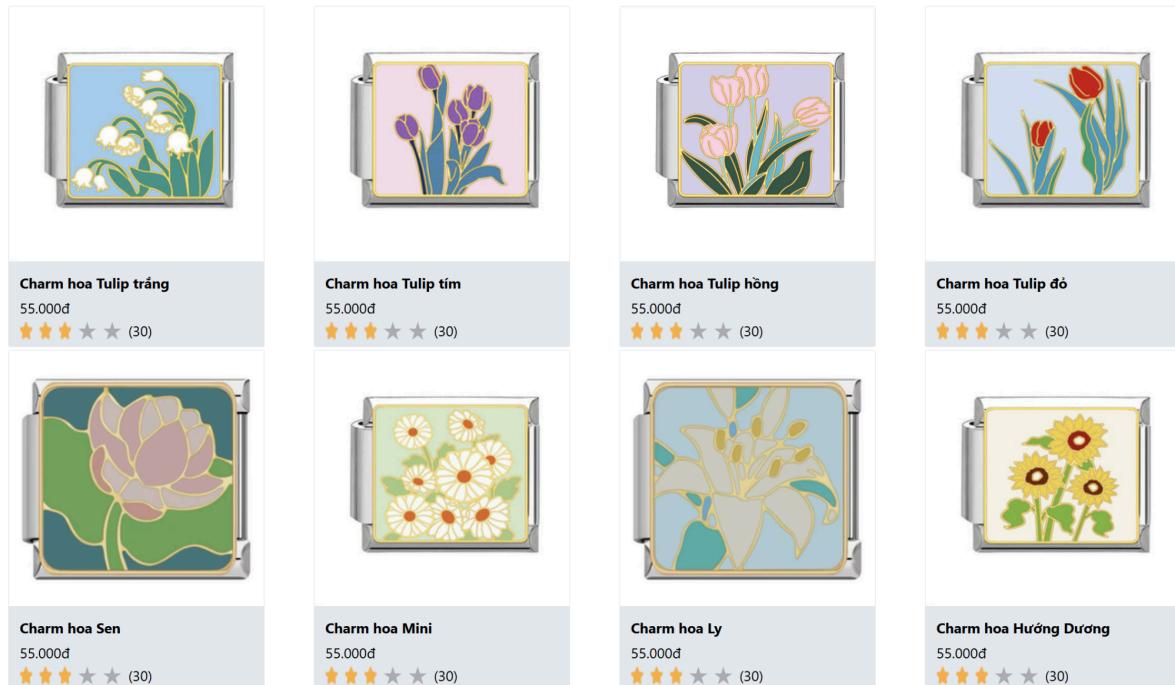
Phần hiển thị sản phẩm chính

Phần hiển thị sản phẩm chính được trình bày dưới dạng lưới sản phẩm, sản phẩm được hiển thị theo dạng lưới với 4 cột.

Mỗi ô sản phẩm bao gồm:

- Hình ảnh charm
- Tên sản phẩm
- Giá sản phẩm
- Đánh giá sản phẩm

TẤT CẢ SẢN PHẨM

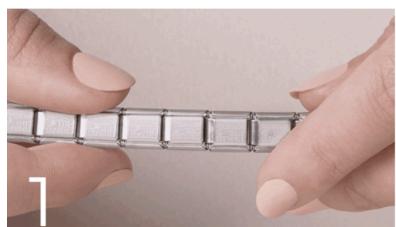


Hình - Lưới sản phẩm chính

Cách sử dụng

Được trình bày dưới dạng ảnh và mô tả ngắn gọn ở phía dưới giúp khách hàng có thể tham khảo được cách sử dụng sản phẩm Charms hay vòng tay của Usbibracelet.

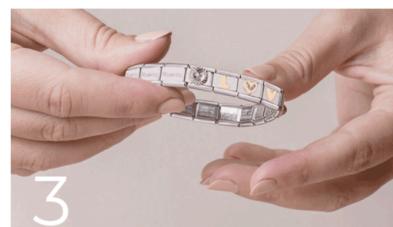
HƯỚNG DẪN SỬ DỤNG



Mở rộng mỗi nốt của vòng đeo tay



Nắm bắt hai đầu của charm muốn tách và tháo ra



Chèn và nối các Charms mới lại với nhau

Hình - Hướng dẫn sử dụng

4.2. Chức năng

Người dùng vào trang Sản phẩm để xem và mua sản phẩm:

Kết nối Cơ sở dữ liệu và Khởi tạo Phiên

```
session_start();  
require_once "include/db.inc.php";
```

- session_start(); khởi tạo phiên làm việc (session) để lưu trữ thông tin người dùng như tên đăng nhập, gio hàng, v.v.
- require_once "include/db.inc.php"; là nơi kết nối đến cơ sở dữ liệu (có thể chứa thông tin cấu hình kết nối như PDO hoặc mysqli).

Câu Truy Văn SQL Lấy Sản Phẩm

```
SELECT  
    p.id AS product_id,  
    p.name AS product_name,  
    p.description,  
    p.quantity,  
    pp.price AS product_price,  
    c.name AS category_name,  
    i.path AS image_path  
FROM product p  
LEFT JOIN (  
    SELECT product_id, price  
    FROM productprice  
    WHERE starting_timestamp = (  
        SELECT MIN(starting_timestamp)  
        FROM productprice pp2  
        WHERE pp2.product_id = productprice.product_id  
    )  
) pp ON p.id = pp.product_id  
LEFT JOIN (  
    SELECT product_id, path  
    FROM image i  
    WHERE id = (  
        SELECT MIN(id)
```

```

        FROM image i2
        WHERE i2.product_id = i.product_id
    )
) i ON p.id = i.product_id
LEFT JOIN category c ON p.category_id = c.id
";

```

- Câu truy vấn SQL này lấy thông tin sản phẩm từ bảng product, productprice, image, và category.
- Sử dụng LEFT JOIN để kết hợp thông tin giá, hình ảnh và danh mục sản phẩm.
- MIN(starting_timestamp) được dùng để lấy giá trị đầu tiên trong bảng productprice.

Thêm Điều Kiện (WHERE, SORT, LIMIT, OFFSET)

Danh mục sản phẩm (categoryId):

```

if ($categoryId !== null) {
    $sql .= " WHERE p.category_id = :categoryId ";
}

```

- Nếu có categoryId, sẽ thêm điều kiện để lọc sản phẩm theo danh mục.

Tìm kiếm (searchTerm):

```

if ($searchTerm !== null) {
    if ($categoryId !== null) {
        $sql .= " AND p.name LIKE :searchTerm ";
    } else {
        $sql .= " WHERE p.name LIKE :searchTerm ";
    }
}

```

- Nếu có từ khóa tìm kiếm (searchTerm), sẽ lọc sản phẩm theo tên sản phẩm.

Sắp xếp (sort):

```

if ($sort === 'asc') {
    $sql .= "ORDER BY pp.price ASC ";
} elseif ($sort === 'desc') {
    $sql .= "ORDER BY pp.price DESC ";
} else {
    $sql .= "ORDER BY p.id ASC "; // Sắp xếp theo ID nếu không có sắp xếp
}

```

- Nếu có giá trị sắp xếp (theo giá tăng dần hoặc giảm dần), câu truy vấn sẽ thay đổi theo điều kiện.

Chuẩn Bị và Thực Thi Truy Vấn

```
$stmt = $pdo->prepare($sql);
// Bind giá trị nếu có categoryId
if ($categoryId !== null) {
    $stmt->bindParam(':categoryId', $categoryId, PDO::PARAM_INT);
}
// Bind giá trị nếu có searchTerm
if ($searchTerm !== null) {
    $stmt->bindValue(':searchTerm', '%' . $searchTerm . '%', PDO::PARAM_STR);
}
// Bind LIMIT và OFFSET cho phân trang
$stmt->bindValue(':limit', $productsPerPage, PDO::PARAM_INT);
$stmt->bindValue(':offset', $offset, PDO::PARAM_INT);
$stmt->execute();
$products = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

- Các tham số được gắn vào câu truy vấn với bindParam và bindValue.
- PDO::PARAM_INT và PDO::PARAM_STR chỉ định kiểu dữ liệu cho các tham số (số nguyên và chuỗi).

Lấy Tổng Số Sản Phẩm

```
$countSql = "
SELECT COUNT(*) AS total FROM product p
LEFT JOIN category c ON p.category_id = c.id
";
if ($categoryId !== null) {
    $countSql .= " WHERE p.category_id = :categoryId ";
}
if ($searchTerm !== null) {
    if ($categoryId !== null) {
        $countSql .= " AND p.name LIKE :searchTerm ";
    } else {
        $countSql .= " WHERE p.name LIKE :searchTerm ";}}
```

- COUNT(*) được dùng để đếm tổng số sản phẩm trong cơ sở dữ liệu.

Lấy Tên Danh Mục

```
$categoryName = "TẤT CẢ SẢN PHẨM";
if ($categoryId !== null) {
```

```

$stmtCategory = $pdo->prepare("SELECT name FROM category WHERE id = :categoryId");
$stmtCategory->bindParam(':categoryId', $categoryId, PDO::PARAM_INT);
$stmtCategory->execute();
$category = $stmtCategory->fetch(PDO::FETCH_ASSOC);
if ($category) {
    $categoryName = $category['name'];
}

```

5. Chi tiết sản phẩm

5.1. Giao diện

Charm hoa Tulip trắng

★★★★★ ★ (30)

55.000đ

- 1 +

THÊM VÀO GIỎ HÀNG

MUA NGAY

PRODUCT DETAILS

SIZE GUIDE

SHIPPING & RETURNS

SIMILAR PRODUCTS

Charm hoa Tulip trắng 55.000đ ★★★★★ ★ (30)	Charm hoa Tulip tím 55.000đ ★★★★★ ★ (30)	Charm hoa Tulip hồng 55.000đ ★★★★★ ★ (30)	Charm hoa Tulip đỏ 55.000đ ★★★★★ ★ (30)	Charm hoa Sen 55.000đ ★★★★★ ★ (30)

Hình - Trang chi tiết sản phẩm

Các thành phần hiển thị từ trên xuống dưới:

- Mục hình ảnh sản phẩm
- Mũi tên chuyển qua hình ảnh sản phẩm tiếp theo
- Mục tên sản phẩm

- Mục sao đánh giá sản phẩm
- Mục điều chỉnh số lượng sản phẩm
- Mục thêm vào giỏ hàng
- Mục Mua ngay
- Mục Product Detail
- Mục Size Guide
- Mục Shipping & Returns
- Mục Similar Products
- Mục Sản Phẩm khác

Các nút tương tác

- **Mũi tên**

Mô tả: Khi người dùng ấn vào biểu tượng mũi tên, sẽ chuyển sang hình ảnh kế tiếp của sản phẩm.

Hiệu ứng: Không có.

Sự kiện: Khi người dùng nhấn vào biểu tượng lập tức chuyển qua hình ảnh khác của sản phẩm.



- **Mục thêm vào giỏ hàng**

Mô tả: Khi người dùng ấn vào mục giỏ hàng, sản phẩm sẽ được thêm vào giỏ hàng

Hiệu ứng: Không có

Sự kiện: Khi người dùng ấn vào mục giỏ hàng, sản phẩm sẽ được lập tức thêm vào giỏ hàng và hiện ở mục giỏ hàng.

THÊM VÀO GIỎ HÀNG

- **Mục mua ngay**

Mô tả: Khi người dùng ấn vào mục mua ngay, đó là đường dẫn link tới trang thanh toán

Hiệu ứng: Không có

Sự kiện: Khi người dùng ấn vào mục mua ngay, lập tức sẽ được điều phối tới trang thanh toán

MUA NGAY

- **Mục Product Detail**

Mô tả: Khi người dùng nhấp vào mục "Product Detail", nội dung chi tiết sẽ được hiển thị.

Hiệu ứng: Không có

Sự kiện: Khi người dùng nhấp vào mục "Product Detail", nội dung chi tiết sẽ lập tức được hiển thị.

PRODUCT DETAILS

+

- **Mục Size Guide**

Mô tả: Khi người dùng nhấp vào mục "Size Guide", nội dung chi tiết sẽ được hiển thị.

Hiệu ứng: Không có

Sự kiện: Khi người dùng nhấp vào mục "Size Guide", nội dung chi tiết sẽ lập tức được hiển thị.

SIZE GUIDE

+

- **Mục Shipping & Returns**

Mô tả: Khi người dùng nhấp vào mục "Shipping & Returns", nội dung chi tiết sẽ được hiển thị.

Hiệu ứng: Không có

Sự kiện: Khi người dùng nhấp vào mục "Shipping & Returns", nội dung chi tiết sẽ lập tức được hiển thị.

SHIPPING & RETURNS

+

- **Mục Sản Phẩm khác**

Mô tả: Khi nhấp vào những sản phẩm liên quan, là đường dẫn link đến trang sản phẩm chi tiết của sản phẩm khác.

Hiệu ứng: Không có

Sự kiện: Khi nhấp vào những sản phẩm liên quan, lập tức được sẽ được điều hướng đến trang chi tiết sản phẩm liên quan.



Hình - Similar Products trang Chi Tiết Sản Phẩm

Tạo slider sản phẩm liên quan

```
document.addEventListener("DOMContentLoaded", function () {
  new Splide("#image-carousel", {
    type: "loop",
   perPage: 5,
    perMove: 1,
  }).mount();
});
```

Mở và đóng

Khi người dùng nhấp vào khu vực product-details-container, đoạn mã sẽ:

- Tìm product-details để kiểm tra xem nó có lớp hidden hay không:
 - Nếu có: Hiển thị chi tiết sản phẩm bằng cách gỡ lớp hidden.
 - Nếu không: Ẩn bằng cách thêm lớp hidden.
- Thay đổi biểu tượng + hoặc - dựa trên trạng thái hiển thị.

Mở và đóng Product Details

```
document
  .getElementById("product-details-container")
  .addEventListener("click", function () {
    var productDetails = document.getElementById("product-details");
```

```
var productDetailsIconTrigger = document.getElementById("product-details-icon-trigger");
);
if (productDetails.classList.contains("hidden")) {
    productDetails.classList.remove("hidden");
    productDetailsIconTrigger.innerText = "-";
} else {
    productDetails.classList.add("hidden");
    productDetailsIconTrigger.innerText = "+";
}
});
```

Mở và đóng Size Guide

```
document
.getElementById("size-guide-container")
.addEventListener("click", function () {
    var sizeGuide = document.getElementById("size-guide");
    var sizeGuideIconTrigger = document.getElementById("size-guide-icon-trigger");
);
if (sizeGuide.classList.contains("hidden")) {
    sizeGuide.classList.remove("hidden");
    sizeGuideIconTrigger.innerText = "-";
} else {
    sizeGuide.classList.add("hidden");
    sizeGuideIconTrigger.innerText = "+";
}
});
```

Mở và đóng phần "Shipping & Return"

```
document
.getElementById("shipping-return-container")
.addEventListener("click", function () {
```

```
var shippingReturn = document.getElementById("shipping-return");
var shippingReturnIconTrigger = document.getElementById(
    "shipping-return-icon-trigger"
);
if(shippingReturn.classList.contains("hidden")) {
    shippingReturn.classList.remove("hidden");
    shippingReturnIconTrigger.innerText = "-";
} else {
    shippingReturn.classList.add("hidden");
    shippingReturnIconTrigger.innerText = "+";
}
});
```

Điều chỉnh số lượng sản phẩm

Khi người dùng nhấn nút +:

- Tăng biến số lượng: Biến quantity tăng thêm 1.
- Cập nhật giao diện: Số lượng hiển thị (quantityElement) tăng lên, ví dụ từ 1 thành 2.
- Ghi vào input ẩn: Giá trị quantityInput.value được cập nhật, dùng để gửi dữ liệu khi thêm vào giỏ hàng.

Khi người dùng nhấn nút -:

- Giảm biến số lượng: Biến quantity giảm đi 1, nhưng không giảm dưới 1.
- Cập nhật giao diện: Số lượng hiển thị (quantityElement) giảm, ví dụ từ 2 thành.
- Ghi vào input ẩn: Giá trị quantityInput.value cũng được cập nhật tương ứng.

```
let quantity = 1;
const quantityElement = document.getElementById("quantity");
const quantityInput = document.getElementById("quantityInput");
const decreaseButton = document.getElementById("decrease-quantity");
const increaseButton = document.getElementById("increase-quantity");
```

```

decreaseButton.addEventListener("click", function () {
    if (quantity > 1) {
        quantity--;
        quantityElement.textContent = quantity;
        quantityInput.value = quantity;
    }
});

increaseButton.addEventListener("click", function () {
    quantity++;
    quantityElement.textContent = quantity;
    quantityInput.value = quantity;
});

```

5.2. Chức năng

Bước 1: Khi người dùng vào trang chi tiết sản phẩm (ví dụ: [product-detail.php?productId=123](#)), hệ thống sẽ lấy thông tin chi tiết của sản phẩm từ cơ sở dữ liệu và hiển thị cho người dùng, bao gồm tên sản phẩm, mô tả, hình ảnh, giá, và các sản phẩm tương tự.

```

$productId = isset($_GET['productId']) ? intval($_GET['productId']) : null;
// Truy vấn thông tin sản phẩm
$stmt = $pdo->prepare("SELECT * FROM product WHERE id = :id");
$stmt->bindParam(':id', $productId, PDO::PARAM_INT);
$stmt->execute();
$product = $stmt->fetch(PDO::FETCH_ASSOC);

$stmtImages = $pdo->prepare("SELECT path FROM image WHERE product_id = :productId");
$stmtImages->bindParam(':productId', $productId, PDO::PARAM_INT);
$stmtImages->execute();
$images = $stmtImages->fetchAll(PDO::FETCH_ASSOC);

```

Bước 2: Người dùng có thể thay đổi số lượng sản phẩm muốn mua (tăng/giảm số lượng) ngay trên trang này. Chức năng này có thể được xử lý bằng JavaScript trên trang hoặc qua form gửi POST.

```
let quantity = 1;

const quantityElement = document.getElementById("quantity");
const quantityInput = document.getElementById("quantityInput");
const decreaseButton = document.getElementById("decrease-quantity");
const increaseButton = document.getElementById("increase-quantity");

decreaseButton.addEventListener("click", function () {
    if (quantity > 1) {
        quantity--;
        quantityElement.textContent = quantity;
        quantityInput.value = quantity;
    }
});

increaseButton.addEventListener("click", function () {
    quantity++;
    quantityElement.textContent = quantity;
    quantityInput.value = quantity;
});
```

Bước 3: Khi người dùng nhấn nút "Thêm vào giỏ hàng" (ở trong trang product-detail.php), form sẽ gửi dữ liệu productId (ID sản phẩm) và quantity (số lượng) đến file add-to-cart.inc.php qua phương thức POST.

```
<form class="col-span-2" method="post" action="include/add-to-cart.inc.php">
    <input type="hidden" name="productId" value="<?php echo $productId ?>" />
    <input type="hidden" name="quantity" id="quantityInput" value="1" />
    <button
        type="submit">
        THÊM VÀO GIỎ HÀNG
    </button>
</form>
```

Khi người dùng nhấn **submit**, dữ liệu sản phẩm sẽ được gửi đi đến **add-to-cart.inc.php**.

Bước 4: Khởi tạo session và lưu lại URL trang hiện tại vào `$_SESSION['previous_page']` để người dùng có thể quay lại trang trước sau khi thao tác xong.

```
session_start();
$_SESSION['previous_page'] = $_SERVER['HTTP_REFERER'];
```

Bước 5: Lấy thông tin sản phẩm và số lượng từ form (gửi qua `$_POST`):

```
$productId = $_POST['productId'];
$quantity = $_POST['quantity'];
```

Bước 6: Khởi tạo giỏ hàng trong session nếu chưa có:

```
if (!isset($_SESSION['cart'])) {
    $_SESSION['cart'] = [];
}
```

Bước 7: Kiểm tra xem sản phẩm đã có trong giỏ hàng chưa:

- Duyệt qua tất cả các sản phẩm trong giỏ hàng (`$_SESSION['cart']`) để tìm sản phẩm có `productId` trùng với `productId` người dùng gửi đến.
- Nếu đã có, tăng số lượng sản phẩm lên theo số lượng người dùng chọn.
- Nếu chưa có, thêm sản phẩm mới vào giỏ.

```
$productExists = false;
foreach ($_SESSION['cart'] as &$item) {
    if ($item['productId'] == $productId) {
        $item['quantity'] += $quantity;
        $productExists = true;
        break;
    }
}
```

Bước 8: Nếu sản phẩm chưa có trong giỏ hàng, thêm sản phẩm vào giỏ với số lượng đã chọn:

```
if (!$productExists) {
    $_SESSION['cart'][] = [
```

```

'productId' => $productId,
'quantity' => $quantity
];
}

```

Bước 9: Đánh dấu rằng sản phẩm đã được thêm vào giỏ hàng (để có thể thông báo cho người dùng):

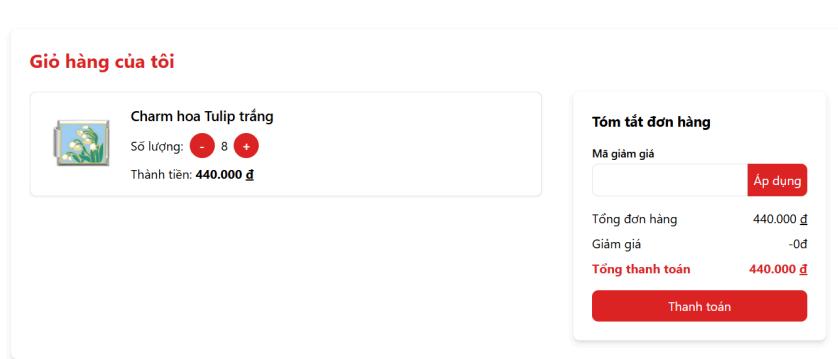
```
$_SESSION['cart_added'] = true;
```

Bước 10: Chuyển hướng về trang trước đó (dùng `$_SESSION['previous_page']` để lấy lại URL của trang trước khi người dùng nhấn thêm vào giỏ hàng).

```
header('Location: ' . $_SESSION['previous_page']);
exit;
```

6. Giỏ hàng

6.1. Giao diện



Hình - Giỏ hàng của tôi

Các thành phần từ trái sang, từ trên xuống dưới

- Hình ảnh sản phẩm
- Tên sản phẩm
- Số lượng sản phẩm
- Nút thêm bớt sản phẩm
- Thành tiền
- Tóm tắt đơn hàng
- Mã giảm giá

- Nút Áp dụng
- Tăng đơn hàng
- Giảm giá
- Tống thanh toán
- Nút Thanh toán

Các tương tác

- **Nút thêm bớt sản phẩm**

Mô tả: Khi người dùng nhấn vào nút + hoặc - , số lượng sản phẩm sẽ tăng hoặc giảm 1 đơn vị.

Hiệu ứng: Không có

Sự kiện: Khi người dùng nhấn vào nút + hoặc - , lập tức số lượng sản phẩm sẽ tăng hoặc giảm 1 đơn vị.



- **Nút Áp dụng**

Mô tả: Khi người dùng nhấn vào nút áp dụng, là đường link tới trang thanh toán.

Hiệu ứng: Không có

Sự kiện: Khi người dùng nhấn vào nút áp dụng, sẽ được điều hướng đến trang thanh toán.



- **Nút Thanh toán**

Mô tả: Khi người dùng nhấn vào nút thanh toán, là đường link tới trang thanh toán.

Hiệu ứng: Không có

Sự kiện: Khi người dùng nhấn vào nút thanh toán, sẽ được điều hướng đến trang thanh toán.

6.2. Chức năng

Bước 1: Khởi tạo và kết nối cơ sở dữ liệu

```
session_start();  
require_once "include/db.inc.php";  
$total = 0;
```

- **session_start();**: Mở phiên làm việc của người dùng, sử dụng để lưu trữ dữ liệu (như giỏ hàng) trong session.
- **require_once "include/db.inc.php";**: Kết nối đến cơ sở dữ liệu thông qua tệp db.inc.php. Trong tệp này, chúng ta sẽ kết nối tới cơ sở dữ liệu MySQL và thiết lập đối tượng PDO để thực hiện truy vấn.
- **\$total = 0;**: Khởi tạo biến \$total để lưu tổng số tiền của giỏ hàng.

Bước 2: Kiểm tra giỏ hàng trong session

```
if (isset($_SESSION['cart'])) {  
    $cartItems = [];  
    foreach ($_SESSION['cart'] as $item) {  
        $productId = $item['productId'];  
        $quantity = $item['quantity'];
```

- **if (isset(\$_SESSION['cart']))**: Kiểm tra xem giỏ hàng đã tồn tại trong session chưa. Nếu có, chúng ta sẽ tiếp tục xử lý giỏ hàng. Nếu không có, mã sau không được thực thi.
- **\$cartItems = [];**: Khởi tạo một mảng rỗng để lưu trữ thông tin các sản phẩm trong giỏ hàng.
- **foreach (\$_SESSION['cart'] as \$item)**: Duyệt qua từng sản phẩm trong giỏ hàng (\$_SESSION['cart']). Mỗi phần tử \$item sẽ chứa thông tin về một sản phẩm, bao gồm productId (ID sản phẩm) và quantity (số lượng).

Bước 3: Truy vấn dữ liệu sản phẩm từ cơ sở dữ liệu

```
$stmt = $pdo->prepare("SELECT * FROM product WHERE id = :id");  
$stmt->bindParam(':id', $productId, PDO::PARAM_INT);  
$stmt->execute();  
$product = $stmt->fetch(PDO::FETCH_ASSOC);
```

- **\$stmt = \$pdo->prepare("SELECT * FROM product WHERE id = :id");**: Truy vấn thông tin sản phẩm từ bảng `product` dựa trên `productId` (ID sản phẩm).
- **\$stmt->bindParam(':id', \$productId, PDO::PARAM_INT);**: Liên kết tham số `:id` với giá trị của `$productId` để bảo vệ chống SQL Injection.
- **\$stmt->execute();**: Thực thi truy vấn.
- **\$product = \$stmt->fetch(PDO::FETCH_ASSOC);**: Lấy dữ liệu của sản phẩm dưới dạng mảng liên kết, giúp truy cập thông qua tên cột.

Bước 4: Kiểm tra và lấy thông tin hình ảnh và giá của sản phẩm

```
if ($product) { // Kiểm tra xem sản phẩm có tồn tại
    $stmtImages = $pdo->prepare("SELECT path FROM image WHERE product_id = :productId");
    $stmtImages->bindParam(':productId', $productId, PDO::PARAM_INT);
    $stmtImages->execute();
    $images = $stmtImages->fetchAll(PDO::FETCH_ASSOC);

    $stmtPrice = $pdo->prepare('SELECT * FROM productprice WHERE product_id = :productId');
    $stmtPrice->bindParam(':productId', $productId, PDO::PARAM_INT);
    $stmtPrice->execute();
    $price = $stmtPrice->fetch(PDO::FETCH_ASSOC);
```

Lấy thông tin hình ảnh:

- Truy vấn bảng `image` để lấy đường dẫn hình ảnh của sản phẩm từ `product_id`.
- **\$stmtImages->fetchAll(PDO::FETCH_ASSOC);**: Lấy tất cả các hình ảnh (nếu có nhiều hình ảnh) dưới dạng mảng liên kết.

Lấy giá sản phẩm:

- Truy vấn bảng `productprice` để lấy giá của sản phẩm.
- **\$stmtPrice->fetch(PDO::FETCH_ASSOC);**: Lấy giá của sản phẩm dưới dạng mảng liên kết.

Bước 5: Xử lý dữ liệu và tính tổng tiền

```
if ($price) { // Kiểm tra xem giá có tồn tại
    // Kết hợp thông tin
    $cartItems[] = [
        'productId' => $product['id'],
        'productName' => $product['name'],
        'productQuantity' => $product['quantity'],
        'quantity' => $quantity,
        'images' => $images,
        'price' => $price['price'], // Thay 'price' bằng tên cột tương ứng nếu cần
    ];
    $total += $quantity * $price['price'];
}
```

Kiểm tra giá: Kiểm tra xem giá sản phẩm có tồn tại trong bảng `productprice` hay không. Nếu có, tiếp tục xử lý.

Lưu thông tin sản phẩm vào `$cartItems[]`:

- Lưu thông tin của sản phẩm vào mảng `$cartItems` bao gồm:
 - ID sản phẩm (`productId`)
 - Tên sản phẩm (`productName`)
 - Số lượng còn lại trong kho (`productQuantity`)
 - Số lượng người dùng yêu cầu (`quantity`)
 - Hình ảnh của sản phẩm (`images`)
 - Giá của sản phẩm (`price`)

Tính tổng tiền: Cập nhật tổng tiền của giỏ hàng bằng cách cộng dồn giá trị của từng sản phẩm (`$quantity * $price['price']`).

Bước 6: Hiển thị giỏ hàng trong HTML

```
<form class="w-[70%] mt-10 mx-auto bg-white p-6 rounded-lg shadow-md"
action="include/checkout.inc.php" method="post">
    <h1 class="text-2xl font-bold text-red-600 mb-6">Giỏ hàng của tôi</h1>
    <div class="grid grid-cols-1 md:grid-cols-3 gap-6">
        <div class="md:col-span-2 space-y-6 max-h-[500px] overflow-auto pr-4">
```

```
<?php if (!empty($cartItems)): ?>
<?php foreach ($cartItems as $item): ?>
<div
    class="flex items-center justify-between p-4 border rounded-lg bg-white
shadow-sm"
>

<div class="flex-1 ml-4">
    <h2 class="text-lg font-semibold"><?= $item['productName'] ?></h2>
    <div class="flex items-center mt-2">
        <span class="mr-2"> Số lượng: </span>
        <button
            class="bg-red-600 font-bold text-white rounded-full w-8 h-8 flex
items-center justify-center btn-decrease"
            data-product-id="<?= $item['productId']; ?>"
            type="button"
>
        <p>-</p>
    </button>
        <span class="mx-2" id="quantity-<?= $item['productId']; ?>"> <?=
$item['quantity'] ?> </span>
        <input
            type="hidden"
            name="quantities[<?= $item['productId']; ?>]"
            id="hidden-quantity-<?= $item['productId']; ?>"
            value="<?= $item['quantity']; ?>"
        />
        <button
```

```

    class="bg-red-600 font-bold text-white rounded-full w-8 h-8 flex
items-center justify-center btn-increase"
        data-product-id="= $item['productId']; ?"
        type="button"
        >
        <p>+</p>
    </button>
</div>
<p class="mt-2">
    Thành tiền:
    <span id="sub-total-= $item['productId']; ?" class="font-bold">
        <?php echo number_format($item['quantity'] * $item['price'], 0, ',', '.'); ?>
    <span class="underline">đ</span>
        </span>
    </p>
    <p class="hidden" id="unit-price-= $item['productId']; "&gt;"&gt;
        &lt;?= $item['price']; ?&gt;
    &lt;/p&gt;
&lt;/div&gt;
&lt;button class="text-gray-500 hover:text-red-600"&gt;
    &lt;i class="fas fa-trash-alt"&gt;&lt;/i&gt;
&lt;/button&gt;
&lt;/div&gt;
&lt;?php endforeach; ?&gt;
&lt;?php else: ?&gt;
    &lt;p&gt;Không có sản phẩm trong giỏ hàng.&lt;/p&gt;
&lt;?php endif; ?&gt;
&lt;/div&gt;
    &lt!-- Thêm thông tin tóm tắt đơn hàng ở đây --&gt;
&lt;/div&gt;
&lt;/form&gt;
</pre

```

Hiển thị danh sách sản phẩm:

- Sử dụng vòng lặp `foreach` để duyệt qua các sản phẩm trong giỏ hàng và hiển thị thông tin chi tiết: tên, hình ảnh, số lượng và thành tiền.
- Thêm các nút `+` và `-` để thay đổi số lượng sản phẩm trong giỏ.

Thông tin thành tiền của từng sản phẩm:

- Số tiền của mỗi sản phẩm được tính toán và hiển thị dưới dạng `number_format` để dễ đọc (ví dụ: `10,000 đ`).

Bước 7: Cập nhật giỏ hàng và tính tổng thanh toán (JavaScript)

```
document.querySelectorAll('.btn-increase').forEach(button => {
    button.addEventListener('click', function () {
        const productId = this.getAttribute('data-product-id');
        const quantitySpan = document.getElementById(`quantity-${productId}`);
        const hiddenInput = document.getElementById(`hidden-quantity-${productId}`);
        const unitPrice = parseFloat(document.getElementById(`unit-price-${productId}`).textContent);
        const subTotalSpan = document.getElementById(`sub-total-${productId}`);
        let quantity = parseInt(quantitySpan.textContent);

        // Tăng số lượng
        quantity += 1;
        quantitySpan.textContent = quantity;
        hiddenInput.value = quantity; // Cập nhật giá trị input ẩn

        // Cập nhật thành tiền
        const subTotal = quantity * unitPrice;
        subTotalSpan.textContent = subTotal.toLocaleString('vi-VN', {
            style: 'currency',
            currency: 'VND'
        });

        // Cập nhật tổng thanh toán
        updateTotal();
    })
})
```

```

    });
}

```

Cập nhật số lượng: Khi người dùng nhấn nút tăng hoặc giảm số lượng sản phẩm, JavaScript sẽ cập nhật số lượng và tổng tiền của sản phẩm tương ứng.

Cập nhật lại giá trị giỏ hàng: Mỗi khi số lượng thay đổi, tổng tiền của sản phẩm được tính lại và cập nhật trực tiếp trên giao diện mà không cần phải tải lại trang.

7. Thanh toán

7.1. Giao diện

7.1.1 Trang thanh toán mục vận chuyển

Tóm tắt đơn hàng		Thông tin liên hệ		Thông tin giao hàng	
	Charm hoa Tulip trắng - 2 +	110.000đ	Tên	Họ	Tỉnh/Thành phố
	Charm hoa Tulip hồng - 1 +	55.000đ	Email		Quận/Huyện
	Charm hoa Tulip tím - 1 +	55.000đ	+ 84	Số điện thoại	Phường/Xã
Mã giảm giá <input type="text" value="Nhập mã giảm giá"/> <input type="button" value="Áp dụng"/>		Địa chỉ nhà <input type="checkbox"/> Lưu địa chỉ của tôi cho lần thanh toán tiếp theo.			
Tổng đơn hàng Phi vận chuyển Giảm giá Tổng thanh toán		220.000đ FREE -0đ 220.000đ	<input type="button" value="Tiếp tục"/>		

Hình - Giao diện trang thanh toán mục vận chuyển

Các chi tiết chính của giao diện gồm:

a. **Tóm tắt đơn hàng (bên trái màn hình):**

Thông tin sản phẩm: Tên sản phẩm, Số lượng và Giá sản phẩm

Ô nhập **Mã giảm giá** và nút "Áp dụng" để xác nhận.

Mã giảm giá

HBTMX10

Áp dụng

Tính toán chi phí: Tổng đơn hàng, Phí vận chuyển, Giảm giá và Tổng thanh toán

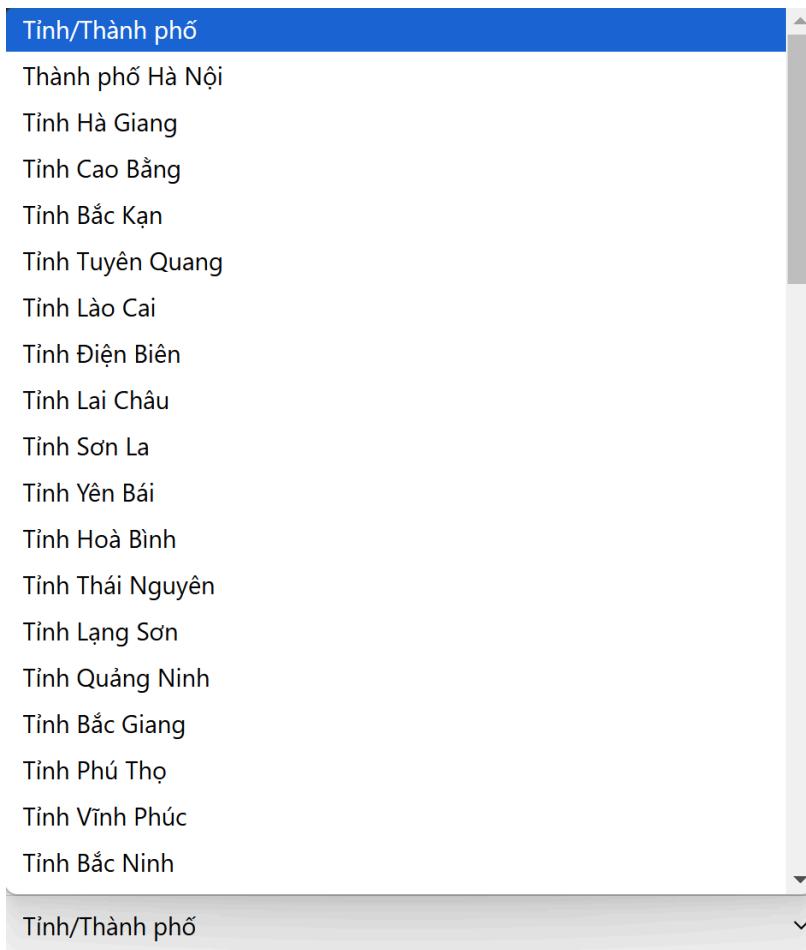
b. **Thông tin liên hệ (bên phải màn hình phía trên):**

Form nhập thông tin của khách hàng gồm các ô nhập **Tên, Họ, Email, Số điện thoại** (với mã quốc gia mặc định là +84).

c. Thông tin địa chỉ giao hàng (bên phải màn hình phía dưới):

Nhóm đã sử dụng đoạn mã JavaScript để xây dựng một hệ thống động cho phép người dùng chọn tỉnh/thành phố, sau đó là quận/huyện, và cuối cùng là phường/xã. Hệ thống này được hỗ trợ dựa trên dữ liệu lấy từ API công khai của trang provinces.open-api.vn.

Ô chọn **Tỉnh/Thành phố** (chọn từ menu thả xuống)



Hình - Ô chọn Tỉnh/Thành phố

Tải danh sách Tỉnh/Thành phố:

```
const provinceSelect = document.getElementById("province-select");

fetch("https://provinces.open-api.vn/api/p/")

  .then((res) => res.json())

  .then((data) => {

    console.log(data);

    data.forEach((province) => {

      const option = document.createElement("option");

      option.value = province.code;
```

```
option.textContent = province.name;  
  
document.getElementById("province-select").appendChild(option);  
  
});  
  
});
```

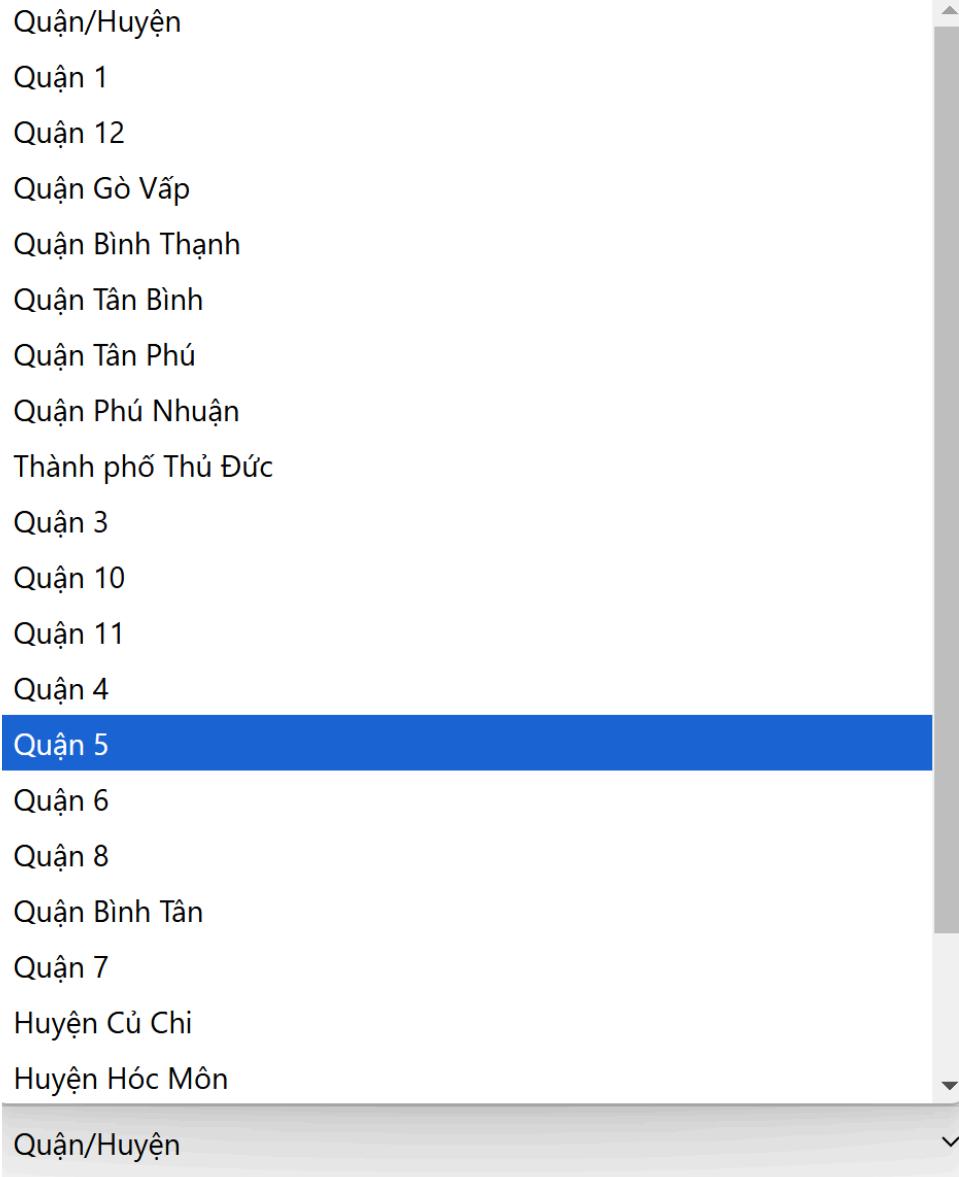
fetch được sử dụng để lấy dữ liệu từ API chứa danh sách tỉnh/thành phố.

Dữ liệu trả về là danh sách các tỉnh/thành phố (JSON), được duyệt qua bằng phương thức forEach.

Với mỗi tỉnh/thành phố, một <option> được tạo và thêm vào <select> với ID là province-select.

Khách hàng có thể chọn tỉnh/thành phố từ danh sách này.

Ô chọn **Quận/Huyện**



Tải danh sách Quận/Huyện dựa trên Tỉnh/Thành phố được chọn

```
const districtSelect = document.getElementById("district-select");

provinceSelect.addEventListener("change", (e) => {

    fetch(`https://provinces.open-api.vn/api/p/${e.target.value}?depth=2`)

    .then((res) => res.json())

    .then((data) => {
```

```
districtSelect.innerHTML = `<option  
value="0">Quận/Huyện</option>`;  
  
data.districts.forEach((district) => {  
  
    const option = document.createElement("option");  
  
    option.value = district.code;  
  
    option.textContent = district.name;  
  
    document.getElementById("district-select").appendChild(option);  
  
});  
  
});  
  
});
```

provinceSelect lắng nghe sự kiện change (khi người dùng chọn một tỉnh/thành phố). Khi người dùng chọn một tỉnh/thành phố, đoạn mã fetch gọi API chi tiết của tỉnh/thành phố đó (dựa trên e.target.value, tức mã tỉnh).

Dữ liệu trả về bao gồm danh sách các quận/huyện thuộc tỉnh/thành phố. Phần này được duyệt qua và tạo ra các <option> cho danh sách quận/huyện (districtSelect).

Trước khi thêm danh sách mới, đoạn mã làm sạch danh sách cũ bằng cách đặt lại innerHTML.

Ô chọn Phường/Xã

Phường/Xã

Phường 15

Phường 13

Phường 14

Phường 12

Phường 11

Phường 10

Phường 09

Phường 01

Phường 08

Phường 02

Phường 04

Phường 07

Phường 05

Phường 06

Phường/Xã

Tải danh sách Phường/Xã dựa trên Quận/Huyện được chọn

```
const wardSelect = document.getElementById("ward-select");

districtSelect.addEventListener("change", (e) => {

    fetch(`https://provinces.open-api.vn/api/d/${e.target.value}?depth=2`)

        .then((res) => res.json())

        .then((data) => {

            wardSelect.innerHTML = `<option value="0">Phường/Xã</option>`;

            data.wards.forEach((ward) => {
```

```
const option = document.createElement("option");

option.value = ward.code;

option.textContent = ward.name;

document.getElementById("ward-select").appendChild(option);

});

});

});
```

districtSelect lắng nghe sự kiện change (khi người dùng chọn một quận/huyện). Khi một quận/huyện được chọn, đoạn mã gọi API chi tiết của quận/huyện đó để lấy danh sách phường/xã (dựa trên e.target.value, tức mã quận/huyện). Dữ liệu trả về được sử dụng để tạo danh sách các <option> cho phường/xã trong wardSelect. Giống như ở bước trước, danh sách cũ sẽ bị xóa trước khi danh sách mới được thêm vào. Ô nhập Địa chỉ nhà (ô nhập văn bản).

522 Nguyễn Tri Phương

Tùy chọn: **Lưu địa chỉ của tôi cho lần thanh toán tiếp theo.**

Lưu địa chỉ của tôi cho lần thanh toán tiếp theo.

d. Nút "Tiếp tục"

Nút “Tiếp tục” màu đỏ ở dưới cùng bên phải để người dùng tiến hành bước thanh toán tiếp theo (di chuyển sang trang thanh toán mục giao hàng)

7.1.2 Trang thanh toán mục giao hàng

The screenshot shows the checkout process on the Usbibracelet website. At the top, there's a navigation bar with links for 'Trang chủ', 'Cửa hàng', 'Tin tức', 'Về chúng tôi', 'Liên hệ', 'Giỏ hàng' (Cart), and 'Đăng xuất' (Logout). A search bar is also present.

The main content area displays a summary of the order:

Tóm tắt đơn hàng	Vận chuyển	Giao hàng	Thanh toán
Charm hoa Tulip trắng	Tu sáp xếp	FREE	
Charm hoa Tulip hồng	Grab giao hàng	+ 20.000đ	
Charm hoa Tulip tím	Bee giao hàng	+ 25.000đ	

Below the summary, there's a section for 'Mã giảm giá' (Discount code) with an input field containing 'HBTMX10' and a red 'Áp dụng' (Apply) button.

The total amount shown is **Tổng thanh toán: 240.000đ**.

At the bottom, there's a footer with links for 'Usbibracelet' (Đăng ký, Nhận ngay mã giảm giá 12%), 'Hỗ trợ' (Đường CMT8, Quận 10, TP HCM, Usbibracelet@gmail.com, 08358588484), 'Menu' (Trang chủ, Bài viết, Cửa hàng, Cửa chuyên Usbibracelet, Giỏ hàng), and 'Theo dõi Usbibracelet'.

Hình - Giao diện trang thanh toán mục giao hàng

Các chi tiết chính của giao diện gồm:

a. Tóm tắt đơn hàng (bên trái màn hình):

Thông tin sản phẩm: Tên sản phẩm, Số lượng và Giá sản phẩm

Ô nhập **Mã giảm giá** và nút "Áp dụng" để xác nhận.

Mã giảm giá

HBTMX10

Áp dụng

Tính toán chi phí: Tổng đơn hàng, Phí vận chuyển, Giảm giá và Tổng thanh toán

b. Phương thức vận chuyển (bên phải màn hình)

Các tùy chọn vận chuyển (chỉ được chọn duy nhất một trong các phương thức): Tự sáp xếp, Grab giao hàng, Bee giao hàng.

Phương thức vận chuyển

- Tự sắp xếp FREE
- Grab giao hàng + 20 000đ
- Bee giao hàng + 25 000đ

[Quay lại](#)

[Tiếp tục](#)

Hình - Phương thức vận chuyển

Để thực hiện chức năng này nhóm đã thực hiện đoạn mã JavaScript gồm 3 phần như sau:

```
<script>
    const doityourself = document.getElementById("doityourself");
    const grab = document.getElementById("grab");
    const bee = document.getElementById("bee");
```

Ba dòng mã này truy xuất các phần tử HTML (radio button hoặc checkbox) dựa trên id. Chúng đại diện cho ba phương thức vận chuyển khác nhau.

Năm bắt sự kiện click của ba phương thức:

Thứ nhất, năm bắt sự kiện click của "doityourself"

```
doityourself.addEventListener("click", () => {
    grab.checked = false;
    bee.checked = false;
    shippingFee.textContent = "FREE";
    shippingFee.classList.add("text-green-500");
    document.getElementById("total").innerHTML = "<?php echo
number_format($total, 0, ',', '.') . 'đ'; ?>";
});
```

Khi người dùng chọn phương thức tự vận chuyển thì đồng nghĩa với các phương thức khác ("grab" và "bee") bị bỏ chọn.

Thứ hai, nắm bắt sự kiện click của "grab"

```
grab.addEventListener("click", () => {
    doityourself.checked = false;
    bee.checked = false;
    shippingFee.textContent = "+ 20 000đ";
    shippingFee.classList.remove("text-green-500");
    document.getElementById("total").innerHTML = "<?php echo
number_format($total + 20000, 0, ',', '.') . 'đ'; ?>";
});
```

Khi người dùng chọn phương thức vận chuyển bằng grab thì đồng nghĩa với các phương thức khác ("doityourself" và "bee") bị bỏ chọn.

Thứ ba, nắm bắt sự kiện click của "bee"

```
bee.addEventListener("click", () => {
    doityourself.checked = false;
    grab.checked = false;
    shippingFee.textContent = "+ 25 000đ";
    shippingFee.classList.remove("text-green-500");
    document.getElementById("total").innerHTML = "<?php echo
number_format($total + 25000, 0, ',', '.') . 'đ'; ?>";
});
```

Khi người dùng chọn phương thức vận chuyển bằng bee thì đồng nghĩa với các phương thức khác ("doityourself" và "grab") bị bỏ chọn.

Các nút điều hướng

Nút Quay lại: Để trở về bước trước.

Nút Tiếp tục: Để chuyển sang bước thanh toán tiếp theo. (di chuyển sang trang thanh toán mục thanh toán).

7.1.3 Trang thanh toán mục thanh toán

The screenshot shows a payment page with the following details:

Trang chủ	Bài viết	Cửa hàng	Về chúng tôi	Tin tức
Tóm tắt đơn hàng				
	Charm hoa Tulip trắng - 2 +	110.000đ	Vận chuyển — <input checked="" type="checkbox"/> — Giao hàng — <input checked="" type="checkbox"/> — Thanh toán	
	Charm hoa Tulip tím - 1 +	55.000đ	Phương thức thanh toán <input type="radio"/> Thanh toán khi nhận hàng (COD) <input type="radio"/> Chuyển khoản ngân hàng <input type="radio"/> Momo	
	Charm hoa Tulip hồng - 1 +	55.000đ	Quay lại	Xác nhận
Mã giảm giá <input type="text" value="Nhập mã giảm giá"/> Áp dụng				
Tổng đơn hàng 220.000đ Phi vận chuyển 0đ Giảm giá -0đ Tổng thanh toán 220.000đ				

Hình - Giao diện trang thanh toán mục thanh toán

Các chi tiết chính của giao diện gồm:

Tóm tắt đơn hàng (bên trái màn hình):

Thông tin sản phẩm: Tên sản phẩm, Số lượng và Giá sản phẩm

Ô nhập Mã giảm giá và nút "Áp dụng" để xác nhận.

Mã giảm giá

<u><u>HBTMX10</u></u>	Áp dụng
-----------------------	-------------------------

Tính toán chi phí: Tổng đơn hàng, Phi vận chuyển, Giảm giá và Tổng thanh toán

Phương thức thanh toán (bên phải màn hình)

Các tùy chọn thanh toán (chỉ được chọn duy nhất một trong các phương thức): Thanh toán khi nhận hàng (COD), Chuyển khoản ngân hàng, Momo.

Vận chuyển — — Giao hàng — — Thanh toán

Phương thức thanh toán

- Thanh toán khi nhận hàng (COD)
- Chuyển khoản ngân hàng
- Momo

[Quay lại](#)

[Xác nhận](#)

Hình - Các phương thức thanh toán

Mục tiêu chính của chức năng này là đảm bảo chỉ có một phương thức thanh toán được chọn tại một thời điểm. Để thực hiện chức năng này nhóm đã thực hiện đoạn mã JavaScript gồm 3 phần như sau.

```
const cod = document.getElementById("cod");
const bank = document.getElementById("bank");
const momo = document.getElementById("momo");
```

Ba biến cod, bank, và momo được liên kết với các checkbox có id tương ứng trong giao diện, đại diện cho các phương thức thanh toán.

Năm bắt sự kiện click của ba phương thức:

Thứ nhất, năm bắt sự kiện click của "cod"

```
cod.addEventListener("click", () => {
  bank.checked = false;
  momo.checked = false;
});
```

Khi người dùng chọn phương thức thanh toán “cod” thì đồng nghĩa với các phương thức khác (“bank” và “momo”) bị bỏ chọn.

Thứ hai, năm bắt sự kiện click của "bank"

```
bank.addEventListener("click", () => {  
    cod.checked = false;  
    momo.checked = false;  
});
```

Khi người dùng chọn phương thức thanh toán “bank” thì đồng nghĩa với các phương thức khác (“cod” và “momo”) bị bỏ chọn.

Thứ ba, nắm bắt sự kiện click của “momo”

```
momo.addEventListener("click", () => {  
    cod.checked = false;  
    bank.checked = false;  
});
```

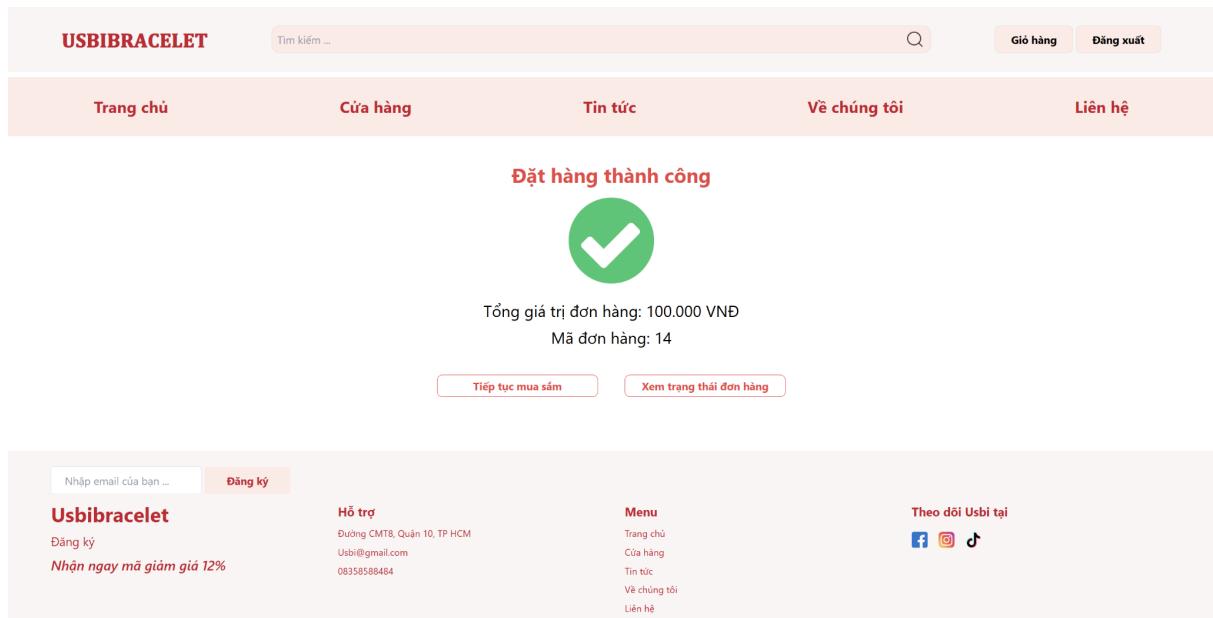
Khi người dùng chọn phương thức thanh toán “momo” thì đồng nghĩa với các phương thức khác (“cod” và “bank”) bị bỏ chọn.

a. Các nút điều hướng

Nút **Quay lại**: Để trở về bước trước.

Nút **Xác nhận**: Để di chuyển sang trang xác nhận đơn hàng.

7.1.4 Giao diện trang “Xác nhận đơn hàng thành công”



Hình - Giao diện trang xác nhận đơn hàng thành công

Sau khi khách hàng hoàn tất bước thanh toán và bấm xác nhận trang web sẽ trả về giao diện xác nhận đơn hàng thành công để thông báo cho người dùng.

7.2. Chức năng

7.2.1 Điene thong tin mua hang

B1: Người dùng sẽ điền thông tin vào form trong file checkout-ship.php

```
<form class="flex justify-between items-start" method="post">
  action="include/checkout-ship.inc.php">
    <!-- Order Summary -->
    <div class="w-1/2 pr-6">
      <h2 class="text-xl font-bold mb-4">Tóm tắt đơn hàng</h2>
      <?php if (!empty($cartItems)): ?>
        <?php foreach ($cartItems as $item): ?>
          <div class="flex items-center mb-4">
            <img
              alt="Product 1 image"
              class="w-16 h-16 mr-4"
              height="60" />
```

```
src="<?php echo $item['images'][0]['path']; ?>"  
width="60"  
/>>  
<div class="flex-1">  
  <p><?= $item['productName'] ?></p>  
  <div class="flex items-center">  
    <button  
      class="border w-6 h-6 flex items-center justify-center font-bold  
rounded-lg bg-slate-200"  
      >  
      -  
    </button>  
    <span class="px-2"><?= $item['quantity'] ?> </span>  
    <button  
      class="border w-6 h-6 flex items-center justify-center font-bold  
rounded-lg bg-slate-200"  
      >  
      +  
    </button>  
  </div>  
</div>  
  <p class="w-20 text-right"><?php echo number_format($item['quantity'] *  
$item['price'], 0, ',', ',') . 'đ'; ?></p>  
  <button class="text-gray-500 ml-4">  
    <i class="fas fa-trash"></i>  
  </button>  
</div>  
<?php endforeach; ?>  
<?php else: ?>  
  <p>Không có sản phẩm trong giỏ hàng.</p>  
<?php endif; ?>  
<div class="mb-4">  
  <label class="block mb-2" for="discount-code"> Mã giảm giá </label>  
<div class="flex">
```

```
<input  
    class="border border-gray-300 p-2 flex-1"  
    id="discount-code"  
    placeholder="Nhập mã giảm giá"  
    type="text"  
/>  
<button class="bg-red-500 text-white px-4 py-2 ml-2">  
    Áp dụng  
</button>  
</div>  
</div>  
<div class="border-t border-gray-300 pt-4">  
    <div class="flex justify-between mb-2">  
        <p>Tổng đơn hàng</p>  
        <p><?php echo number_format($total, 0, ',', '.') . 'đ'; ?></p>  
    </div>  
    <div class="flex justify-between mb-2">  
        <p>Phí vận chuyển</p>  
        <p class="text-green-500">FREE</p>  
    </div>  
    <div class="flex justify-between mb-2">  
        <p>Giảm giá</p>  
        <p>-0đ</p>  
    </div>  
    <div class="flex justify-between font-bold text-red-500">  
        <p>Tổng thanh toán</p>  
        <p><?php echo number_format($total, 0, ',', '.') . 'đ'; ?></p>  
    </div>  
</div>  
</div>  
<!-- Contact and Shipping Information -->  
<div class="w-1/2 pl-6">  
    <div class="mb-5 items-center flex gap-1">  
        <p class="text-red-500 font-semibold">Vận chuyển</p>
```

```
<div class="flex gap-2 items-center">
  <div class="border w-5 translate-y-0.5"></div>
  <div
    class="border flex items-center justify-center font-bold text-white text-xs
translate-y-0.5 h-5 w-5 rounded-full bg-slate-200"
  >
    ✓
  </div>
  <div class="border w-5 translate-y-0.5"></div>
</div>
<p class="font-semibold text-slate-500">Giao hàng</p>
<div class="flex gap-2 items-center">
  <div class="border w-5 translate-y-0.5"></div>
  <div
    class="border flex items-center justify-center font-bold text-white text-xs
translate-y-0.5 h-5 w-5 rounded-full bg-slate-200"
  >
    ✓
  </div>
  <div class="border w-5 translate-y-0.5"></div>
</div>
<p class="font-semibold text-slate-500">Thanh toán</p>
</div>
<div>
  <div class="mb-6">
    <h2 class="text-xl font-bold mb-4">Thông tin liên hệ</h2>
    <div class="grid grid-cols-2 gap-4 mb-4">
      <input
        class="border border-gray-300 p-2"
        placeholder="Tên"
        name="first_name"
        type="text"
      />
      <input
```

```
        class="border border-gray-300 p-2"
        placeholder="Họ"
        name="last_name"
        type="text"
    />
</div>
<input
    class="border border-gray-300 p-2 w-full mb-4"
    placeholder="Email"
    name="email"
    type="email"
/>
<div class="flex items-center mb-4">
    <select class="border border-gray-300 p-2 mr-2">
        <option>+ 84</option>
    </select>
    <input
        class="border border-gray-300 p-2 flex-1"
        placeholder="Số điện thoại"
        name="phoneNumber"
        type="text"
    />
</div>
</div>
<div class="mb-6">
    <h2 class="text-xl font-bold mb-4">Thông tin giao hàng</h2>
    <select
        id="province-select"
        name="provinceId"
        class="border border-gray-300 p-2 w-full mb-4"
    >
        <option value="0">Tỉnh/Thành phố</option>
    </select>
    <select
```

```
        id="district-select"
        name="districtId"
        class="border border-gray-300 p-2 w-full mb-4"
    >
        <option value="0">Quận/Huyện</option>
    </select>
    <select
        id="ward-select"
        name="wardId"
        class="border border-gray-300 p-2 w-full mb-4"
    >
        <option value="0">Phường/Xã</option>
    </select>
    <input
        class="border border-gray-300 p-2 w-full mb-4"
        placeholder="Địa chỉ nhà"
        name="address"
        type="text"
    />
    <div class="flex items-center mb-4">
        <input class="mr-2" id="save-address" type="checkbox" />
        <label for="save-address">
            Lưu địa chỉ của tôi cho lần thanh toán tiếp theo.
        </label>
    </div>
    </div>
    <div class="flex justify-end">
        <button id="countinue"
            type="submit"
            class="bg-red-500 text-white px-6 py-2 w-32"
        >
            Tiếp tục
        </button>
    </div>
```

```
</div>
</div>
</form>
```

B2: Data sẽ được gửi về file checkout-ship.inc.php

```
($_SERVER["REQUEST_METHOD"] == "POST") {
    $first_name = $_POST["first_name"];
    $last_name = $_POST["last_name"];
    $email = $_POST["email"];
    $phoneNumber = $_POST["phoneNumber"];
    $provinceId = $_POST["provinceId"];
    $districtId = $_POST["districtId"];
    $wardId = $_POST["wardId"];
    $address = $_POST["address"];
    $account_id = $_SESSION["user_id"];
```

Các biến này lấy dữ liệu từ form đặt hàng (gửi qua POST). Đây thường là thông tin như họ tên, số điện thoại, địa chỉ giao hàng.

`$_SESSION["user_id"]`: ID của người dùng hiện tại (đã đăng nhập). Nếu `user_id` không tồn tại, người dùng được chuyển hướng đến trang đăng nhập.

```
if ($account_id == null) {
    header("Location: ../login.php");
    exit();
}
```

Nếu người dùng chưa đăng nhập (`$account_id == null`), họ sẽ được chuyển đến trang đăng nhập để tiếp tục.

```
$cartItems = [];

if (isset($_SESSION['cart'])) {
    foreach ($_SESSION['cart'] as $item) {
        $productId = $item['productId'];
        $quantity = $item['quantity'];
```

`$_SESSION['cart']`: Giỏ hàng được lưu trữ trong session. Mỗi sản phẩm có:

- `productId`: ID sản phẩm.
- `quantity`: Số lượng người dùng chọn.

```
$stmt = $pdo->prepare("SELECT * FROM product WHERE id = :id");
$stmt->bindParam(':id', $productId, PDO::PARAM_INT);
$stmt->execute();
$product = $stmt->fetch(PDO::FETCH_ASSOC);
```

Truy vấn sản phẩm dựa trên productId. Lấy thông tin sản phẩm từ bảng product.

```
$stmtImages = $pdo->prepare("SELECT path FROM image WHERE product_id = :productId");
$stmtPrice = $pdo->prepare('SELECT * FROM productprice WHERE product_id = :productId');
```

Lấy ảnh sản phẩm từ bảng image và giá từ bảng productprice.

```
$total += $quantity * $price['price'];
```

Tính tổng giá tiền giờ hàng bằng cách nhân số lượng sản phẩm với giá sản phẩm.

```
$queryOrder = "INSERT INTO `order` (account_id, status, phonenumber,
provinceId, districtId, wardId, address, total)

VALUE (?, ?, ?, ?, ?, ?, ?, ?, ?);";

$stmtOrder = $pdo->prepare($queryOrder);

$stmtOrder->execute([$account_id, "pending", $phoneNumber, $provinceId,
$districtId, $wardId, $address, $total]);

$orderId = $pdo->lastInsertId();

$_SESSION['order_id'] = $orderId;
```

- Lưu thông tin đơn hàng (người dùng, trạng thái, địa chỉ, tổng tiền) vào bảng order.
- Lấy ID của đơn hàng vừa tạo bằng \$pdo->lastInsertId().

```

$queryOrderDetail = "INSERT INTO `orderdetail` (product_id, quantity, price,
order_id)

        VALUE (?, ?, ?, ?);"

$stmtOrderDetail = $pdo->prepare($queryOrderDetail);

        $stmtOrderDetail->execute([$cartItem['productId'], $cartItem['quantity'],
$cartItem['price'], $orderId]);

```

Thêm từng sản phẩm trong giỏ hàng vào bảng orderdetail, lưu các thông tin như:

- product_id: ID sản phẩm.
- quantity: Số lượng.
- price: Giá.
- order_id: ID đơn hàng.

7.2.2 Chọn phương thức vận chuyển

B1: Khách hàng chọn phương thức giao hàng.

```

<form method="post" action="include/checkout-shippingMethod.inc.php">

    <div class="border rounded mb-5 px-4 py-2 shadow-lg">
        <h1 class="font-bold text-lg">Phương thức vận chuyển</h1>
        <div class="mt-3">
            <div class="flex items-center justify-between">
                <div class="flex gap-5">
                    <input id="doityourself" value="doityourself" name="shippingMethod"
type="radio" class="bg-red-500" />
                    <label for="doityourself" class="-translate-y-0.5">Tự sắp xếp</label>
                </div>
                <p class="text-green-400">FREE</p>
            </div>
            <div class="flex mt-1 items-center justify-between">
                <div class="flex gap-5">
                    <input id="grab" value="grab" type="radio" name="shippingMethod"
class="bg-red-500" />
                    <label for="grab" class="-translate-y-0.5">Grab giao hàng</label>
                </div>
            </div>
        </div>
    </div>
</form>

```

```

    </div>
    <p class="">+ 20 000đ</p>
</div>
<div class="flex mt-1 items-center justify-between">
    <div class="flex gap-5">
        <input id="bee" type="radio" value="bee" name="shippingMethod"
class="bg-red-500" />
        <label for="bee" class="-translate-y-0.5">Bee giao hàng</label>
    </div>
    <p class="">+ 25 000đ</p>
    </div>
</div>
</div>
<div class="flex justify-end gap-7">
    <button
        id="back"
        type="button"
        class="border border-red-500 text-red-500 px-6 py-2 w-32"
    >
        Quay lại
    </button>
    <button
        type="submit"
        class="bg-red-500 text-white px-6 py-2 w-32"
    >
        Tiếp tục
    </button>
</div>
</form>

```

B2: Data sẽ được gửi về file checkout-shippingMethod.inc.php

```

$shippingMethod = $_POST["shippingMethod"];
if ($_SESSION['order_id'] == null) {
    header("Location: ../login.php");
    exit();
}

```

```
}
```

\$shippingMethod: Lấy phương thức vận chuyển từ form gửi lên (ví dụ: Grab hoặc Bee).

Kiểm tra xem \$_SESSION['order_id'] có tồn tại không:

- Nếu không có đơn hàng trong session (tức là người dùng chưa đặt hàng), họ sẽ bị chuyển hướng đến trang đăng nhập.

```
$shippingFee = 0;  
if ($shippingMethod == 'grab') {  
    $shippingFee = 20000;  
} else if ($shippingMethod == 'bee') {  
    $shippingFee = 25000;  
}
```

\$shippingFee: Xác định phí vận chuyển tùy theo phương thức vận chuyển

Lưu vào Session:

```
$_SESSION['shippingMethod'] = $shippingMethod;
```

```
$_SESSION['shippingFee'] = $shippingFee;
```

```
$queryOrder = "SELECT * FROM `order` WHERE id = :id";
```

```
$stmtOrder = $pdo->prepare($queryOrder);  
$stmtOrder->bindParam(":id", $_SESSION['order_id']);  
$stmtOrder->execute();
```

```
$result = $stmtOrder->fetch(PDO::FETCH_ASSOC);
```

```
$total = $result['total'];
```

- queryOrder: Truy vấn thông tin đơn hàng hiện tại dựa vào order_id (lấy từ session).
- Lấy tổng giá trị đơn hàng hiện tại từ cột total trong bảng order.

```
if ($result) {  
    if ($result['shippingMethod'] == 'grab') {  
        $shippingFee -= 20000;  
    } else if ($result['shippingMethod'] == 'bee') {  
        $shippingFee -= 25000;  
    }
```

```
$total += $shippingFee;
```

Nếu đơn hàng đã có một phương thức vận chuyển trước đó:

- Giảm phí vận chuyển tương ứng (trừ đi 20,000 hoặc 25,000).
- Sau đó, cộng lại phí vận chuyển mới vào tổng giá trị đơn hàng.

```
$updateOrder = "UPDATE `order` SET shippingMethod = :shippingMethod, total = :total WHERE id = :id;";
$stmtOrder = $pdo->prepare($updateOrder);
$stmtOrder->bindParam(":id", $_SESSION['order_id']);
$stmtOrder->bindParam(":shippingMethod", $shippingMethod);
$stmtOrder->bindParam(":total", $total);
$stmtOrder->execute();
```

updateOrder: Cập nhật:

- shippingMethod: Phương thức vận chuyển mới.
- total: Tổng tiền (bao gồm cả phí vận chuyển).
- Dựa vào order_id trong session.

7.2.3 Chọn phương thức thanh toán

B1: Khách hàng chọn phương thức thanh toán trong file checkout-paymentMethod.php

```
<form action="include/checkout-paymentMethod.inc.php" method="post">
    <div class="border rounded mb-5 px-4 py-2 shadow-lg">
        <h1 class="font-bold text-lg">Phương thức vận chuyển</h1>
        <div class="mt-3">
            <div class="flex items-center justify-between">
                <div class="flex gap-5">
                    <input id="cod" value="cod" name="paymentMethod" type="radio" class="bg-red-500" />
                    <label for="cod" class="-translate-y-0.5">Thanh toán khi nhận hàng (COD)</label>
                </div>
            </div>
            <div class="flex mt-1 items-center justify-between">
                <div class="flex gap-5">
```

```

        <input id="bank" value="bank" type="radio" name="paymentMethod"
class="bg-red-500" />
            <label for="bank" class="-translate-y-0.5">Chuyển khoản ngân
hàng</label>
        </div>
    </div>
    <div class="flex mt-1 items-center justify-between">
        <div class="flex gap-5">
            <input id="momo" value="momo" type="radio" name="paymentMethod"
class="bg-red-500" />
                <label for="momo" class="-translate-y-0.5">Momo</label>
            </div>
        </div>
        </div>
    </div>
    <div class="flex justify-end gap-7">
        <button
            id="back"
            type="button"
            class="border border-red-500 text-red-500 px-6 py-2 w-32"
>
            Quay lại
        </button>
        <button
            type="submit"
            class="bg-red-500 text-white px-6 py-2 w-32"
>
            Xác nhận
        </button>
    </div>
</form>

```

B2: Data sẽ gửi về file checkout-paymentMethod.inc.php

```

if($_SESSION['order_id'] == null) {
    header("Location: ../login.php");
}

```

```
exit();
```

Mục đích: Đảm bảo rằng đơn hàng (order_id) tồn tại trong session.

Vai trò:

- Nếu không có order_id (có thể do người dùng chưa đăng nhập hoặc chưa đặt hàng), họ sẽ được chuyển hướng về trang đăng nhập (../login.php).
- Lệnh exit() dừng thực thi script ngay sau khi chuyển hướng.

```
require_once "db.inc.php";
```

Mục đích: Kết nối đến cơ sở dữ liệu bằng cách sử dụng tệp db.inc.php.

Vai trò: Đảm bảo script có thể thực hiện các thao tác với cơ sở dữ liệu.

```
$updateOrder = "UPDATE `order` SET paymentMethod = :paymentMethod, status = :status WHERE id = :id;";  
$stmtOrder = $pdo->prepare($updateOrder);  
$stmtOrder->bindParam(":id", $_SESSION['order_id']);  
$stmtOrder->bindParam(":paymentMethod", $paymentMethod);  
$stmtOrder->bindParam(":status", $status);  
$stmtOrder->execute();
```

Mục đích: Cập nhật phương thức thanh toán và trạng thái đơn hàng trong bảng order.

Chi tiết:

- \$paymentMethod được lấy từ form POST (dữ liệu người dùng gửi lên).
- status được cố định là "successed", thể hiện trạng thái đơn hàng đã hoàn tất.
- order_id được lấy từ session (\$_SESSION['order_id']).

Cách hoạt động:

- SQL: Lệnh UPDATE dùng để cập nhật thông tin trong bảng order.
- Prepared Statement:
 - + Giúp bảo mật chống lại SQL Injection.
 - + Dữ liệu được gán vào câu truy vấn thông qua bindParam.

7.2.4 Thông báo thanh toán thành công

Sau khi thanh toán hoàn tất khách hàng sẽ được chuyển tiếp sang trang thông báo đơn hàng thành công.

```
<div class="flex flex-col items-center mt-10">  
  <h1 class="text-red-500 font-bold text-3xl">Đặt hàng thành công</h1>  
  <h3 class="mt-4 text-8xl">✓</h3>
```

```

<h3 class="mt-6 text-2xl">Tổng giá trị đơn hàng: 100.000 VNĐ</h3>
<h3 class="mt-2 text-2xl">Mã đơn hàng: <?= $_SESSION['order_id'] ?></h3>
<div class="mt-10 grid grid-cols-2 gap-10">
  <button
    class="w-60 py-1 text-red-500 font-bold border border-red-500 rounded-lg">
    Tiếp tục mua sắm
  </button>
  <button
    class="w-60 py-1 text-red-500 font-bold border border-red-500 rounded-lg">
    Xem trạng thái đơn hàng
  </button>
</div>
</div>

```

8. Lọc sản phẩm

8.1 Giao diện

Sắp xếp theo

[Charm cho nữ](#)

[Charm cho nam](#)

[Charm lủng lẳng](#)

[Charm chữ số](#)

[Sample mix sản](#)

[Charm dài](#)

[Charm đính đá](#)

Charm cho nữ



Charm hoa Tulip trắng
55.000đ
 (30)



Charm hoa Tulip tím
55.000đ
 (30)



Charm hoa Tulip hồng
55.000đ
 (30)



Charm hoa Tulip đỏ
55.000đ
 (30)

Các thành phần hiển thị phần lọc sản phẩm từ trái sang:

- Lọc Charm cho nữ
- Lọc Charm cho nam
- Lọc Charm chữ số
- Lọc Charm lủng lẳng
- Lọc Charm đính đá
- Lọc Charm dài
- Lọc Sample mix sản

- Tất cả sản phẩm đã được lọc

Các tương tác:

- Mô tả: Là nút để người dùng chọn sản phẩm mà họ mong muốn.
- Hiệu ứng: Xuất hiện một đoạn thăng gạch dưới tên mục mình chọn khi người dùng ấn chuột vào.
- Sự kiện: Hiển thị danh sách sản phẩm mà người dùng đã chọn.

8.2 Chức năng

Bước 1: Người dùng chọn các bộ lọc trên giao diện (Frontend)

- Người dùng có thể thực hiện các hành động lọc thông qua giao diện:
 - **Chọn danh mục:** Chọn danh mục sản phẩm từ một danh sách.
 - **Tìm kiếm theo tên sản phẩm:** Nhập từ khóa tìm kiếm vào ô tìm kiếm.
 - **Sắp xếp theo giá:** Chọn sắp xếp sản phẩm theo giá (tăng dần hoặc giảm dần).

Ví dụ, nếu người dùng muốn xem sản phẩm của một danh mục cụ thể, họ sẽ chọn danh mục từ dropdown. Nếu họ muốn tìm kiếm, họ sẽ nhập từ khóa vào ô tìm kiếm.

Bước 2: Gửi tham số lọc lên server (Client-side - JavaScript)

- Sau khi người dùng thực hiện thao tác lọc, **JavaScript** sẽ lấy giá trị của các bộ lọc và gửi nó lên server thông qua URL. Thông thường, tham số được gửi qua URL dưới dạng query string (ví dụ: `?categoryId=1&sort=asc&searchTerm=shoes`).

Ví dụ với dropdown sắp xếp:

Khi người dùng thay đổi lựa chọn trong dropdown sắp xếp (`#sort`), sự kiện `change` sẽ được kích hoạt, cập nhật URL và làm mới trang.

```
btn_login = document.getElementById("btn_login");
```

```
btn_login.addEventListener("click", function () {
```

```
window.location.href = "/login.php";  
});  
  
document.getElementById("product-detail").addEventListener("click", function ()  
{  
  
    window.location.href = "/product-detail.php?product_id=1";  
});  
  
// Chỉ cho phép chọn một category  
  
document.addEventListener('DOMContentLoaded', function() {  
  
    const sortElement = document.getElementById('sort');  
  
    if (sortElement) {  
  
        sortElement.addEventListener('change', function() {  
  
            const sortValue = this.value;  
  
            const url = new URL(window.location.href);  
  
            if (sortValue !== 'default') {  
  
                url.searchParams.set('sort', sortValue); // Thêm tham số sort vào URL  
  
            } else {  
  
                url.searchParams.delete('sort'); // Nếu không chọn sắp xếp, xóa tham số sort  
  
            }  
  
            window.location.href = url.toString(); // Làm mới trang với URL mới  
        });  
    }  
});
```

- **Tìm kiếm:** Khi người dùng nhập từ khóa vào ô tìm kiếm, một sự kiện có thể được thêm vào để gửi từ khóa tìm kiếm (`searchTerm`) lên server qua URL. Ví dụ, sau khi người dùng nhấn nút tìm kiếm, URL có thể trông như sau: `?searchTerm=laptop`.
- **Lọc theo danh mục:** Khi người dùng nhấp vào một danh mục, URL sẽ thay đổi theo tham số `categoryId`. Ví dụ: `?categoryId=3`.

Bước 3: Server nhận và xử lý tham số lọc từ URL

Trên server, đoạn mã PHP sẽ nhận các tham số từ URL và sử dụng chúng để tạo ra câu truy vấn SQL tương ứng. Các tham số có thể là:

- `categoryId`: Danh mục sản phẩm.
- `sort`: Phương thức sắp xếp (theo giá, tăng dần hoặc giảm dần).
- `searchTerm`: Từ khóa tìm kiếm sản phẩm.

```
// Kiểm tra nếu có categoryId trên URL
$categoryId = isset($_GET['categoryId']) ? intval($_GET['categoryId']) : null;

// Kiểm tra nếu có sort trên URL
$sort = isset($_GET['sort']) ? $_GET['sort'] : 'default'; // chưa viết docs đoạn này

// Kiểm tra nếu có searchTerm trong URL
$searchTerm = isset($_GET['searchTerm']) ? $_GET['searchTerm'] : null;

// Kiểm tra nếu có trang (page) trong URL
$page = isset($_GET['page']) ? intval($_GET['page']) : 1; // Mặc định là trang 1

// Số lượng sản phẩm trên mỗi trang
$productsPerPage = 12;

// Tính toán offset dựa vào trang hiện tại
$offset = ($page - 1) * $productsPerPage;
```

- **Kiểm tra và sử dụng các tham số:**

- Nếu có `categoryId`, chỉ lọc sản phẩm thuộc danh mục đó.
- Nếu có `searchTerm`, lọc sản phẩm theo tên.
- Nếu có `sort`, sắp xếp sản phẩm theo giá (tăng dần hoặc giảm dần).

Bước 4: Xây dựng câu truy vấn SQL động

Câu truy vấn SQL sẽ được xây dựng dựa trên các tham số đã nhận được từ URL, bao gồm các điều kiện lọc và sắp xếp.

```
// Xây dựng câu truy vấn SQL cơ bản
```

```
$sql = "  
SELECT  
    p.id AS product_id,  
    p.name AS product_name,  
    p.description,  
    p.quantity,  
    pp.price AS product_price,  
    c.name AS category_name,  
    i.path AS image_path  
FROM product p  
LEFT JOIN (  
    SELECT product_id, price  
    FROM productprice  
    WHERE starting_timestamp = (  
        SELECT MIN(starting_timestamp)
```

```
        FROM productprice pp2
        WHERE pp2.product_id = productprice.product_id
    )
) pp ON p.id = pp.product_id
LEFT JOIN (
    SELECT product_id, path
    FROM image i
    WHERE id =
        SELECT MIN(id)
    FROM image i2
    WHERE i2.product_id = i.product_id
)
) i ON p.id = i.product_id
LEFT JOIN category c ON p.category_id = c.id
";
// Thêm điều kiện WHERE nếu có categoryId
if ($categoryId !== null) {
    $sql .= " WHERE p.category_id = :categoryId ";
}
// Thêm điều kiện tìm kiếm nếu có searchTerm
if ($searchTerm !== null) {
```

```
if ($categoryId !== null) {  
  
    $sql .= " AND p.name LIKE :searchTerm ";  
  
} else {  
  
    $sql .= " WHERE p.name LIKE :searchTerm ";  
  
}  
  
}  
  
  
// Thêm điều kiện ORDER BY nếu có sắp xếp  
  
if ($sort === 'asc') {  
  
    $sql .= "ORDER BY pp.price ASC ";  
  
} elseif ($sort === 'desc') {  
  
    $sql .= "ORDER BY pp.price DESC ";  
  
} else {  
  
    $sql .= "ORDER BY p.id ASC "; // Sắp xếp theo ID nếu không có sắp xếp  
  
}  
  
// Thêm LIMIT và OFFSET cho phân trang  
  
$sql .= "LIMIT :limit OFFSET :offset";
```

Câu truy vấn này sẽ trả về danh sách sản phẩm phù hợp với các điều kiện lọc đã được chỉ định. Các kết quả này sẽ được sử dụng để hiển thị lên giao diện.

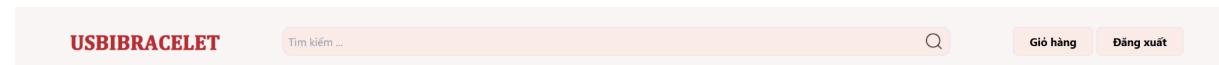
Bước 6: Gửi dữ liệu về giao diện người dùng (Frontend)

Kết quả sản phẩm từ câu truy vấn SQL sẽ được trả về và được xử lý để hiển thị lên giao diện người dùng. Có thể sử dụng `foreach` để duyệt qua các sản phẩm và hiển thị chúng.

```
foreach ($products as $product)
```

9. Tìm kiếm sản phẩm

9.1 Giao diện



Hình - Thanh tìm kiếm sản phẩm

Thanh **Tìm kiếm sản phẩm**: Cho phép người dùng nhập từ khóa liên quan đến sản phẩm họ muốn tìm kiếm, giúp tiết kiệm thời gian khi duyệt website và tăng trải nghiệm mua hàng.

Kích hoạt nhanh kết quả: Với biểu tượng kính lúp, người dùng có thể nhấn nút tìm kiếm để hệ thống hiển thị các kết quả, sản phẩm charm và vòng tay liên quan đến từ khóa mà khách hàng đã nhập.

9.2 Chức năng

Bước 1: Tạo Form Tìm Kiếm trong Header (HTML), nơi người dùng có thể nhập từ khóa tìm kiếm và gửi yêu cầu đến server.

```
<form class="relative flex w-3/5 items-center" action=".//include/search-product.inc.php" method="post"> <input class="w-full rounded-xl border bg-[#FFEAEA] p-2" placeholder="Tìm kiếm ..." name="searchTerm" /> <button class="absolute right-3 h-6">  </button> </form>
```

Bước 2: Xử lý Dữ Liệu Tìm Kiếm trong `search-product.inc.php`

Khi form được gửi, cần xử lý dữ liệu tìm kiếm trong file search-product.inc.php. Đây là nơi thực hiện các kiểm tra và chuyển hướng người dùng đến trang kết quả tìm kiếm.

```
<?php
session_start();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $searchTerm = $_POST["searchTerm"];
    if (isset($searchTerm))
        header("Location: ../product-list.php?searchTerm=". $searchTerm);
} else {
    header("Location: ../index.php");
    die();
}
?>
```

- **`$_POST["searchTerm"]`**: Lấy giá trị từ trường nhập liệu `searchTerm` trong form tìm kiếm.
- **`trim()`**: Xóa khoảng trắng thừa trước và sau từ khóa.**`htmlspecialchars()`**: Làm sạch ký tự đặc biệt để tránh tấn công XSS.
- **`urlencode()`**: Mã hóa từ khóa tìm kiếm trước khi truyền qua URL.
- **`header("Location: ...")`**: Chuyển hướng người dùng đến trang `product-list.php` và gửi từ khóa qua query string (`?searchTerm=...`).

Bước 3: Xử lý và Hiển Thị Kết Quả Tìm Kiếm trong `product-list.php`

Sau khi người dùng được chuyển hướng đến trang `product-list.php`, cần lấy từ khóa tìm kiếm từ URL và sử dụng nó để tìm kiếm trong cơ sở dữ liệu, sau đó hiển thị kết quả.

```
// Kiểm tra nếu có searchTerm trong URL
$searchTerm = isset($_GET['searchTerm']) ? $_GET['searchTerm'] : null;

// Kiểm tra nếu có trang (page) trong URL
```

```
$page = isset($_GET['page']) ? intval($_GET['page']) : 1; // Mặc định là trang 1

// Số lượng sản phẩm trên mỗi trang

$productsPerPage = 12;

// Tính toán offset dựa vào trang hiện tại

$offset = ($page - 1) * $productsPerPage;

// Xây dựng câu truy vấn SQL cơ bản

$sql = "

SELECT

    p.id AS product_id,
    p.name AS product_name,
    p.description,
    p.quantity,
    pp.price AS product_price,
    c.name AS category_name,
    i.path AS image_path

FROM product p

LEFT JOIN (
    SELECT product_id, price
    FROM productprice
    WHERE starting_timestamp = (
        SELECT MIN(starting_timestamp)
        FROM productprice pp2
```

```

        WHERE pp2.product_id = productprice.product_id

    )

) pp ON p.id = pp.product_id

LEFT JOIN (
    SELECT product_id, path
    FROM image i
    WHERE id = (
        SELECT MIN(id)
        FROM image i2
        WHERE i2.product_id = i.product_id
    )
    ) i ON p.id = i.product_id

LEFT JOIN category c ON p.category_id = c.id
";

```

Bước 4: Hiển Thị Kết Quả Tìm Kiếm

Trang `product-list.php` sẽ hiển thị các sản phẩm có tên chứa từ khóa tìm kiếm hoặc thông báo không tìm thấy sản phẩm phù hợp.

```

<div class="col-span-8">
    <p class="text-2xl font-bold text-[#CE112D]"><?= $categoryName ?></p>    <div
    class="mt-3 grid grid-cols-4 gap-x-6 gap-y-1">
        <?php if (empty($products)): ?>
            <p>Không tìm thấy sản phẩm nào</p>

```

```
<?php else: ?>

<?php foreach ($products as $product): ?>

    <div id="product-detail" class="h-96 w-72 rounded-sm border
bg-slate-200">

        <div class="mt-3 px-3">

            <a href="product-detail.php?productId=<?= $product['product_id'] ?>">
                <?= $product['product_name'] ?>
            </a>

            <p class="mt-1"><?php echo number_format($product['product_price'], 0, ',', '.') . 'đ'; ?></p>

            <div class="flex items-center gap-2">

                <div class="mt-1 flex gap-2">

                </div>

                </div>

                <p class="translate-y-0.5">(30)</p>

            </div>

        </div>

    </div>

<?php endforeach; ?>
```

```
<?php endif; ?>
```

```
</div>
```

Bước 5: Hiển thị phân trang

```
<ul class="flex justify-center gap-4">

<?php for ($i = 1; $i <= $totalPages; $i++): ?>

<li>
<a href="?page=<?= $i ?>

    <?php if (!empty($categoryId)) echo '&categoryId=' . $categoryId;
?>

    <?php if (!empty($sort) && $sort !== 'default') echo '&sort=' . $sort;
?>

    <?php if (!empty($searchTerm)) echo '&searchTerm=' .
urlencode($searchTerm); ?>"

        class="px-4 py-2 <?= $i === $page ? 'bg-red-500 text-white' :
'bg-gray-300' ?>">

        <?= $i ?>

    </a>
</li>

<?php endfor; ?>

</ul>
```

10. Tin tức

10.1 Giao diện

10.1.1 Danh sách tin tức



Usbibracelet Ideas

Món quà cưới hoàn hảo cho người bạn đời của bạn

Đám cưới của người bạn thân nhất của bạn là khoảnh khắc xứng đáng nhận được một món quà đặc biệt như dịp này. Đó là cơ hội để bạn thể hiện bản thân trong tình bạn của mình như thế nào và bạn vui mừng như thế nào về chương mới thù vị của họ. Nhưng món quà lý tưởng để thể hiện tất cả những điều này là gì? Câu trả lời rất đơn giản: một món đồ trang sức

→



Trang sức vàng hồng: 5 lựa chọn đeo cùa cho mùa hè

Vàng hồng được yêu thích vì màu sắc ấm áp, langle mạn mang đến nét tinh tế cho bất kỳ trang phục nà...



Trang sức gia đình có biểu tượng: Liên kết tình thân

Tặng đồ trang sức tôn vinh gia đình là một cử chỉ vô cùng ý nghĩa, giàu cảm xúc và ý nghĩa: nó vượt qua...



Đá quý có ý nghĩa gì?

Không chỉ đầy màu sắc và đẹp: nhiều ý nghĩa khác nhau đã được gán cho các loại đá quý tự nhiên đã t...



Ngày của Mẹ: những câu nói hay dành tặng mẹ

Lời nói rất quan trọng; một thông điệp viết ra sẽ không bao giờ phai mờ. Bạn đã bao giờ thấy mình...



20 tuổi: một món quà sinh nhật độc đáo và mang tính cá nhân

Bước sang tuổi 20 là một thời điểm quan trọng trong cuộc đời . Giống như bạn chính thức bước qua...



Quà tặng tốt nghiệp

Tốt nghiệp trung học là một thành tựu quan trọng đối với những người trẻ tuổi. Sau nhiều năm học, là...



Mua gì tặng bố mẹ vào dịp Giáng sinh

Quà tặng cho bố mẹ xứng đáng đóng vai trò chính trong số tất cả những món quà Giáng sinh bạn cần...



Quà tặng cho bạn bè: Ý tưởng tạo ra một món đồ trang sức độc đáo

Không có gì đẹp đẽ và thú vị hơn việc tặng một món quà độc đáo và trên hết là chân thành . Đó là lý do t...

Hình ảnh: Trang danh sách tin tức

Các thành phần hiển thị từ trái sang phải, từ trên xuống dưới. Giao diện gồm 9 bài viết.

- Bài viết 1: Hiển thị hình ảnh bài viết, Tiêu đề bài viết, Nội dung một đoạn bài viết, mũi tên điều hướng đến bài viết.
- Bài viết 2: Hiển thị hình ảnh bài viết, Tiêu đề bài viết, Nội dung một đoạn bài viết.
- Bài viết 3: Hiển thị hình ảnh bài viết, Tiêu đề bài viết, Nội dung một đoạn bài viết.
- Bài viết 4: Hiển thị hình ảnh bài viết, Tiêu đề bài viết, Nội dung một đoạn bài viết.
- Bài viết 5: Hiển thị hình ảnh bài viết, Tiêu đề bài viết, Nội dung một đoạn bài viết.

- Bài viết 6: Hiển thị hình ảnh bài viết, Tiêu đề bài viết, Nội dung một đoạn bài viết.
- Bài viết 7: Hiển thị hình ảnh bài viết, Tiêu đề bài viết, Nội dung một đoạn bài viết.
- Bài viết 8: Hiển thị hình ảnh bài viết, Tiêu đề bài viết, Nội dung một đoạn bài viết
- Bài viết 9: Hiển thị hình ảnh bài viết, Tiêu đề bài viết, Nội dung một đoạn bài viết

Các nút tương tác:

Tiêu đề bài viết

- Mô tả: Tiêu đề bài viết được hiển thị dưới dạng đường dẫn (link), cho phép người dùng nhấn vào để truy cập chi tiết bài viết.
- Hiệu ứng: Phóng to bài viết khi người dùng di chuột qua.
- Sự kiện: Khi người dùng nhấn chọn vào tiêu đề bài viết, lập tức sẽ được điều phối đến trang chi tiết bài viết.

5.2 Chi tiết tin tức

“THE LOVE OF MINE”

QUÀ TẶNG Ý NGHĨA CHO NGÀY QUỐC TẾ NAM GIỚI

Ngày 19 tháng 11 năm 2024



Ngày Quốc Tế Nam Giới là dịp đặc biệt để tôn vinh phái mạnh, ghi nhận những đóng góp to lớn của họ trong gia đình, công việc, và xã hội. Đây cũng là cơ hội để chúng ta bày tỏ sự trân trọng, quan tâm và yêu thương đối với những người đàn ông quan trọng trong cuộc đời mình. Không chỉ là ngày để nhắc nhở về ý nghĩa của sự gắn kết, đây còn là thời điểm lý tưởng để trao đi những món quà thể hiện tình cảm chân thành, và trang sức chắc chắn là một lựa chọn tinh tế và đáng nhớ.



Lịch sử và ý nghĩa của Ngày Quốc Tế Nam Giới

Ngày Quốc Tế Nam Giới ra đời lần đầu vào năm 1999 tại Trinidad và Tobago, được tổ chức vào ngày 19 tháng 11 hàng năm. Mục tiêu của ngày này không chỉ là tôn vinh phái mạnh mà còn nâng cao nhận thức về sức khỏe nam giới, khuyến khích bình đẳng giới, và nhấn mạnh vai trò của những người cha, người anh, người bạn trong gia đình cũng như công đồng.



Trang sức – Món quà lý tưởng cho Ngày Quốc Tế Nam Giới

Trong dịp đặc biệt này, một món trang sức phù hợp sẽ là cách tuyệt vời để thể hiện sự quan tâm và tình yêu dành cho người đàn ông của bạn. Vòng tay khắc thông điệp cá nhân – Vòng tay bằng thép không gỉ hoặc da thật với thiết kế đơn giản nhưng mạnh mẽ là lựa chọn tuyệt vời. Hãy khắc thêm một thông điệp cá nhân hoặc những con số có ý nghĩa để món quà trở nên đặc biệt hơn. Ví dụ: vòng tay khắc tên hoặc ngày kỷ niệm sẽ luôn nhắc nhở anh ấy về mối quan hệ đặc biệt của hai người.



Hình ảnh: Trang chi tiết tin tức

Các thành phần hiển thị từ trên xuống dưới:

- Tiêu đề Blog
- Ngày, tháng, năm đăng Blog
- Nội dung Blog kèm hình ảnh minh họa

10.2. Chức năng

10.2.1 Xem tổng quát trang Blog

Người dùng sẽ xem các bài viết trực tiếp trên trang bài viết của Webiste Usbibracelet.

- Lấy dữ liệu blog từ database

```
session_start();
require_once "include/db.inc.php";
```

- session_start(): Khởi tạo session cho người dùng để lưu thông tin tạm thời khi cần.
- require_once: Gọi file db.inc.php, nơi chứa thông tin kết nối đến cơ sở dữ liệu (\$pdo là đối tượng PDO kết nối).

```
$sql = "
SELECT blog.* , image.path AS image_path
FROM blog
LEFT JOIN image ON blog.id = image.blog_id
LIMIT :offset, :perPage
";
$stmt = $pdo->prepare($sql);
```

- LEFT JOIN: Liên kết bảng blog và image để lấy dữ liệu blog cùng đường dẫn hình ảnh (image.path).
- :offset & :perPage: Giới hạn số lượng blog trả về từ cơ sở dữ liệu dựa trên phân trang.
- fetchAll(PDO::FETCH_ASSOC): Lấy tất cả kết quả trả về dưới dạng mảng liên kết.

```
$totalBlogsSql = "SELECT COUNT(*) FROM blog";
$totalBlogsStmt = $pdo->query($totalBlogsSql);
$totalBlogs = $totalBlogsStmt->fetchColumn();
$totalPages = ceil($totalBlogs / $perPage);
```

- COUNT(*): Đếm tổng số blog có trong bảng blog.
- ceil: Tính tổng số trang cần thiết để hiển thị toàn bộ blog.

10.2.2 Xem chi tiết từng blog

Người dùng có thể chọn xem chi tiết từng bài viết.

```
session_start();
require_once "include/db.inc.php";
```

- require_once "include/db.inc.php"; để kết nối đến cơ sở dữ liệu thông qua tệp db.inc.php.

Kiểm tra và lấy dữ liệu bài viết:

```
if (isset($_GET['blogId'])) {
    $blogId = (int)$_GET['blogId'];
```

- Kiểm tra xem blogId có được truyền qua URL không (dạng example.com?page.php?blogId=123).
- Chuyển giá trị blogId thành kiểu số nguyên để đảm bảo an toàn (tránh tấn công SQL Injection).

```
$sql = "SELECT * FROM blog WHERE id = :blogId";
$stmt = $pdo->prepare($sql);
$stmt->bindParam(':blogId', $blogId, PDO::PARAM_INT);
$stmt->execute();
```

- Truy vấn bảng blog trong cơ sở dữ liệu để lấy bài viết có id trùng với blogId.
- Sử dụng PDO (PHP Data Object) và câu lệnh prepare() để bảo vệ khỏi SQL Injection.

```
if ($stmt->rowCount() > 0) {
    $blog = $stmt->fetch(PDO::FETCH_ASSOC);
} else {
    echo "Bài viết không tồn tại.";
    exit;
}
} else {
    echo "Không có blogId trong URL.";
    exit;
}
```

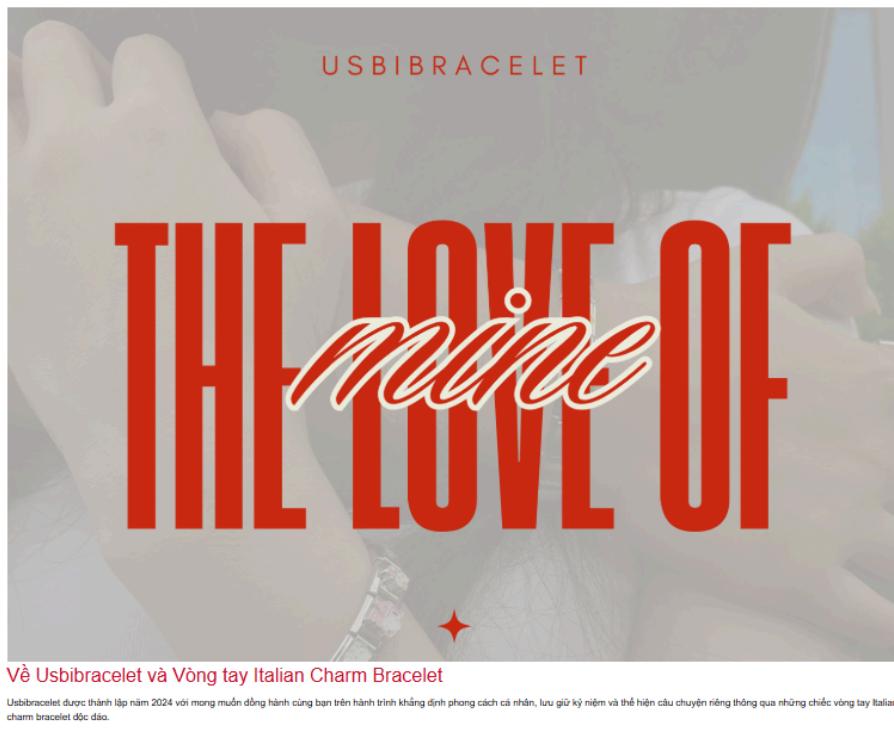
- Nếu bài viết tồn tại, kết quả được lưu vào biến \$blog (lấy dạng mảng kết hợp).
- Nếu không tìm thấy bài viết, thông báo lỗi và thoát.

Xử lý ngày đăng bài:

```
$createdAt = new DateTime($blog['timestamp']);  
$formattedDate = $createdAt->format('d-m-Y H:i:s');
```

- Chuyển đổi trường timestamp (cột trong bảng blog) thành đối tượng DateTime.
- Định dạng ngày giờ thành dd-mm-yyyy hh:mm:ss.

11. Về chúng tôi



Về Usbibracelet và Vòng tay Italian Charm Bracelet

Usbibracelet được thành lập năm 2024 với mong muốn đồng hành cùng bạn trên hành trình khẳng định phong cách cá nhân, lưu giữ kỷ niệm và thể hiện câu chuyện riêng thông qua những chiếc vòng tay Italian charm bracelet độc đáo.

Hình. Trang giới thiệu cửa hàng (1)

Vòng tay nhà Usbibracelet

Sản phẩm của chúng tôi không chỉ là món trang sức lấp lánh, mà còn là cách bạn khắc họa dấu ấn cá nhân, gửi gắm những thông điệp ý nghĩa trong từng chiếc charm.

Với cam kết mang đến những sản phẩm chất lượng cao, đội ngũ Usbibracelet lựa chọn từng chất liệu tinh tế, từ kim loại bền bỉ đến thiết kế charm tinh xảo, đảm bảo mỗi chiếc vòng đều phản ánh sự đặc biệt và giá trị cá nhân của bạn.

Chúng tôi hiểu rằng trang sức không chỉ là phụ kiện, mà còn là cách bạn thể hiện cá tính, cảm xúc và kết nối với những người thân yêu. Với vòng tay Italian charm bracelet từ Usbibracelet, bạn có thể kể nên câu chuyện của riêng mình, ghi dấu ấn và trân trọng khoảnh khắc đáng nhớ và trao gửi từng kỷ niệm.



Các dịch vụ mà Usbibracelet cung cấp

Thiết kế đa dạng

Cung cấp các dòng vòng tay với thiết kế phong phú, từ cổ điển đến hiện đại.

Cá nhân hóa charm

Dịch vụ khắc họa tên, hình ảnh, hoặc thông điệp riêng lên charm.

Set quà tặng ý nghĩa

Tạo các set quà đặc đáo cho các dịp đặc biệt như sinh nhật, lễ kỷ niệm.

Giao hàng nhanh

Giao hàng trong ngày tại TP.HCM và Hà Nội.

Dịch vụ doanh nghiệp

Nhận đặt số lượng lớn làm quà tặng đối tác hoặc nhân viên.

Thiết kế miễn phí

Hỗ trợ thiết kế hộp quà và thiệp tặng miễn phí.

Giá trị mà Usbibracelet hướng đến

- Cá nhân hóa và sáng tạo
- Hiện đại và thân thiện
- Tận tâm và gần gũi với khách hàng
- Lan tỏa yêu thương và ý nghĩa qua từng sản phẩm

Usbibracelet tin rằng hành trình khẳng định phong cách và lưu giữ những khoảnh khắc quý giá cùng khách hàng vẫn còn dài và tràn đầy tiềm năng. Chúng tôi rất mong được sự ủng hộ và đồng hành từ bạn trên hành trình này.

Thân thương,

Usbibracelet - YOUR OWN BRACELET, YOUR OWN STORY

Hình. Trang giới thiệu cửa hàng (2)

Trang giới thiệu cửa hàng gồm:

- Banner của Usbi
- Giới thiệu về cửa hàng và sản phẩm
- Giới thiệu về dịch vụ
- Giá trị mà Usbibracelet hướng đến

12. Liên hệ



LIÊN HỆ USBI

Usbi sẽ phản hồi tin nhắn của bạn trong vòng 24h.

Họ

Tên

Email

Số điện thoại VN - 091-234-5678

Nhấn chọn nút này, bạn đồng ý với [Chính sách Bảo mật](#) của Usbi.

Gửi liên hệ

Hình. Trang liên hệ cửa hàng

12.1 Các thành phần từ trái sang

- Bản đồ với địa chỉ cửa hàng
- Form liên hệ yêu cầu các thông tin: Họ Tên, Email, Số điện thoại, Tin nhắn
- Nút nhấn đồng ý với chính sách bảo mật
- Nút Gửi liên hệ

12.2 Các tương tác

Nút Gửi liên hệ:

- Mô tả: Sau khi người dùng nhập thông tin liên hệ (tên, email, số điện thoại, hoặc nội dung liên hệ), hệ thống sẽ kiểm tra tính hợp lệ của dữ liệu và lưu thông tin vào Database. Nếu thông tin hợp lệ, sẽ hiển thị thông báo “Gửi liên hệ thành công”. Nếu không, sẽ báo lỗi “Thông tin không hợp lệ, vui lòng kiểm tra lại”.

- Hiệu ứng: Đổi màu khi di chuột vào.
- Sự kiện: Gửi thông tin liên hệ vào hệ thống và hiển thị thông báo trạng thái (thành công hoặc thất bại).

CHƯƠNG B: GIAO DIỆN QUẢN LÝ

1. Quản lý người dùng

1.1 Giao diện

The screenshot shows a web-based application interface for managing users. On the left, there is a sidebar with icons and labels: 'USBIBRACELET' (with a logo), 'NGƯỜI DÙNG' (User), 'SẢN PHẨM' (Product), 'ĐƠN HÀNG' (Order), and 'TIN TỨC' (News). The main area has a header 'Người dùng' (User) with a search bar 'Tim kiếm' (Search) and two notification icons. A red button 'Thêm ID mới' (Add new ID) is visible. Below is a table with columns: ID, HỌ (Last Name), TÊN (First Name), EMAIL, VAI TRÒ (Role), and HÀNH ĐỘNG (Actions). One row is shown: ID 2, HỌ Phạm Thị Mỹ, TÊN Tâm, EMAIL account@gmail.com, VAI TRÒ user, and an 'Edit' button.

Hình - Giao diện Quản lý người dùng

1.1.1 Các thành phần từ trái sang phải, từ trên xuống dưới:

- Logo Usbibracelet
- Icon Người dùng
- Icon Sản phẩm
- Icon Đơn hàng
- Icon Tin tức
- Thanh tìm kiếm
- NútThêm ID mới
- ID: Mã số người dùng
- HỌ: Họ người dùng
- TÊN: Tên người dùng
- EMAIL: Email người dùng khi đăng ký
- VAI TRÒ: Là admin hay user
- HÀNH ĐỘNG: Có thể sửa tài khoản người dùng
- Nút Edit

1.1.2 Các tương tác

- Icon Người dùng

Mô tả: Khi người dùng ấn vào ô Người dùng, là đường dẫn tới Trang Quản lý người dùng.

Hiệu ứng: Ô Người dùng chuyển đỏ khi di chuột qua

Sự kiện: Khi người dùng ấn vào ô Người dùng, lập tức được điều hướng đến Trang Quản lý người dùng.



- Icon Sản phẩm

Mô tả: Khi người dùng ấn vào ô Sản phẩm, là đường dẫn tới Trang Quản lý Sản phẩm.

Hiệu ứng: Ô Sản phẩm chuyển đỏ khi di chuột qua

Sự kiện: Khi người dùng ấn vào ô Sản phẩm, lập tức được điều hướng đến Trang Quản lý Sản phẩm.

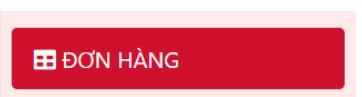


- Icon Đơn hàng

Mô tả: Khi người dùng ấn vào ô Đơn hàng, là đường dẫn tới Trang Quản lý đơn hàng.

Hiệu ứng: Ô Đơn hàng chuyển đỏ khi di chuột qua

Sự kiện: Khi người dùng ấn vào ô Đơn hàng, lập tức được điều hướng đến Trang Quản lý đơn hàng.



- Icon Tin tức

Mô tả: Khi người dùng ấn vào ô Tin tức, là đường dẫn tới Trang Quản lý Tin tức.

Hiệu ứng: Ô Tin tức chuyển đỏ khi di chuột qua

Sự kiện: Khi người dùng ấn vào ô Tin tức, lập tức được điều hướng đến Trang Quản lý Tin tức.

TIN TỨC

- Thanh tìm kiếm

Mô tả: Người dùng có thể tìm kiếm tên người dùng, nếu có sẽ hiển thị trên trang Quản lý người dùng

Hiệu ứng: Không có

Sự kiện: Người dùng ấn vào ô tìm kiếm và nhập tên người dùng muốn tìm kiếm, màn hình trang Quản lý người dùng hiển thị tên người dùng nếu có.

Tìm kiếm

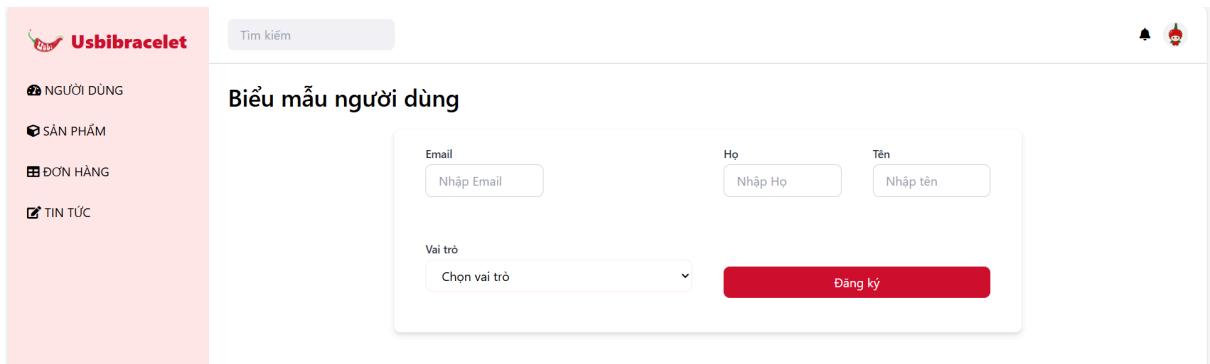
- Nút Thêm ID mới

Mô tả: Người dùng ấn vào nút thêm ID mới, là đường dẫn link tới trang Chi tiết quản lý người dùng

Hiệu ứng: Không có

Sự kiện: Người dùng ấn vào nút thêm ID mới, lập tức sẽ được điều hướng đến trang Chi tiết quản lý người dùng

Thêm ID mới



The screenshot shows a user interface for managing users. On the left, there's a sidebar with navigation links: NGƯỜI DÙNG, SẢN PHẨM, ĐƠN HÀNG, and TIN TỨC. The main area has a header 'Biểu mẫu người dùng'. It contains input fields for 'Email' (Nhập Email), 'Họ' (Nhập Họ), 'Tên' (Nhập tên), and 'Vai trò' (Chọn vai trò). A red 'Đăng ký' button is at the bottom right of the form.

- Nút Edit

Mô tả: Người dùng ấn vào nút Edit, là đường dẫn link tới trang Chi tiết quản lý người dùng

Hiệu ứng: Không có

Sự kiện: Người dùng ấn vào nút Edit, lập tức sẽ được điều hướng đến trang Chi tiết quản lý người dùng

Edit

1.2 Chức năng

1.2.1 User Management

Phía trên danh sách người dùng có nút **Add new user**. Dấn đến trang user-management-detail.php để thêm người dùng mới.

```
<a href="user-management-detail.php">
    <button class="w-40 h-12 bg-slate-300 font-semibold border rounded-lg">Add new user</button>
</a>
```

Admin sẽ xem được thông tin của toàn bộ người dùng

```
<thead class="bg-gray-50">
    <tr>
        <th
            class="py-2 px-4 border-b border-gray-200 text-left text-sm font-semibold text-gray-600">
            ID
        </th>
        <th
            class="py-2 px-4 border-b border-gray-200 text-left text-sm font-semibold text-gray-600">
            FIRST NAME
        </th>
        <th
            class="py-2 px-4 border-b border-gray-200 text-left text-sm font-semibold text-gray-600">
            LAST NAME
        </th>
    </tr>
</thead>
```

```
 EMAIL | ROLE | ACTION |
```

Và Admin có thể chỉnh sửa thông tin người dùng:

```


    <button
        class="rounded-md bg-slate-800 py-2 px-4 border border-transparent text-center text-sm text-white transition-all shadow-md hover:shadow-lg focus:bg-slate-700 focus:shadow-none active:bg-slate-700 hover:bg-slate-700 active:shadow-none disabled:pointer-events-none disabled:opacity-50 disabled:shadow-none ml-2" type="button">
        Edit
    </button>
</a>

```

1.2.2 User management detail

Admin có thể tùy chỉnh người dùng hoặc thêm mới dựa theo file User management detail.php.

```
<form class="" action="include/user-management-detail.inc.php" method="POST">
    <!-- ID Field -->
    <div class="grid grid-cols-2 items-center gap-10">
        <div class="mb-4 w-full">
            <label for="id" class="block text-sm font-medium text-gray-700 mb-1">ID</label>
            <input
                type="text"
                id="id"
                name="id"
                class="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-none focus:ring focus:ring-indigo-200"
                placeholder="Enter ID"
                value="<?= $account['id'] ?>"
            />
        </div><!-- Email Field -->
        <div class="mb-4">
            <label for="email" class="block text-sm font-medium text-gray-700 mb-1">Email</label>
            <input
                type="email"
                id="email"
                name="email"
                class="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-none focus:ring focus:ring-indigo-200"
                placeholder="Enter Email"
                value="<?= $account['email'] ?>"
                required
            />
        </div>
    </div>
```

```
<div class="grid grid-cols-2 items-center gap-10">
    <!-- First Name Field -->
    <div class="mb-4">
        <label for="firstname" class="block text-sm font-medium text-gray-700 mb-1">First Name</label>
        <input
            type="text"
            id="firstname"
            name="firstname"
            class="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-none focus:ring focus:ring-indigo-200"
            placeholder="Enter First Name"
            value=<?= $account['firstname'] ?>
            required
        />
    </div>
    <!-- Last Name Field -->
    <div class="mb-4">
        <label for="lastname" class="block text-sm font-medium text-gray-700 mb-1">Last Name</label>
        <input
            type="text"
            id="lastname"
            name="lastname"
            class="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-none focus:ring focus:ring-indigo-200"
            placeholder="Enter Last Name"
            value=<?= $account['lastname'] ?>
            required
        />
    </div>
</div>
<!-- Role Field -->
<div class="mb-4">
```

```

        <label for="role" class="block text-sm font-medium text-gray-700
mb-1">Role</label>
        <select
            id="role"
            name="role"
            class="w-full px-4 py-2 border border-gray-300 rounded-lg
focus:outline-none focus:ring focus:ring-indigo-200"
            value=<?= $account['role'] ?>">
        >
        <option value="">Select Role</option>
        <option value="admin">Admin</option>
        <option value="user">User</option>
    </select>
</div>
<!-- Submit Button -->
<div class="mt-6">
    <button
        type="submit"
        class="w-full bg-indigo-600 text-white py-2 px-4 rounded-lg
hover:bg-indigo-700 focus:outline-none focus:ring focus:ring-indigo-300">
        >
        Submit
    </button>
</div>
</form>

```

Form quản lý người dùng hiển thị các trường thông tin:

- **ID:** Không bắt buộc phải nhập; thường được điền tự động nếu chỉnh sửa thông tin người dùng. Nếu đang tạo mới một tài khoản, giá trị này có thể để trống (thường ID sẽ được tự động sinh ra trong cơ sở dữ liệu).
- **Email:** Là trường **bắt buộc** vì email là một thông tin quan trọng để định danh tài khoản. Admin có thể thay đổi email của người dùng.
- **First Name** và **Last Name:** Đây là thông tin cá nhân bắt buộc. Admin có thể thay đổi hoặc nhập vào các trường này.

- **Role:** Admin có quyền gán quyền cho người dùng bằng cách chọn **Admin** hoặc **User** từ dropdown menu.

Các trường này được điền tự động nếu có dữ liệu từ \$account.

Sau khi Admin chỉnh sửa hoặc thêm người dùng vào dashboard. Dữ liệu sẽ được gửi về file user-management-detail.inc.php

```
$insertAccount = "INSERT INTO account (email, firstname, lastname, role)  
VALUES (:email, :firstname, :lastname, :role);";
```

- Sử dụng truy vấn INSERT INTO để thêm tài khoản mới.
- Gắn các tham số email, firstname, lastname, và role vào câu lệnh chuẩn bị truy vấn (prepared statement).

```
$updateAccount = "UPDATE account SET firstname = :firstname, lastname =  
:lastname, role = :role WHERE id = :id;";
```

- Sử dụng truy vấn UPDATE để chỉnh sửa thông tin tài khoản hiện có.
- Prepared Statements: Các tham số được bind (bindParam) để chống SQL Injection.

```
header("Location: ../user-management.php");  
$_SESSION["message"] = "Account saved successfully!";  
die();
```

- Điều hướng về trang quản lý người dùng (user-management.php).
- Lưu thông báo thành công vào \$_SESSION["message"].

```
die("Query failed: " . $e->getMessage());
```

Xử lý lỗi: Nếu xảy ra lỗi trong truy vấn, sẽ dừng chương trình và hiển thị thông báo lỗi.

2. Quản lý sản phẩm

2.1 Giao diện

The screenshot shows a web-based application for managing products. On the left, there's a sidebar with icons for 'NGƯỜI DÙNG' (User), 'SẢN PHẨM' (Product), 'ĐƠN HÀNG' (Order), and 'TIN TỨC' (News). The main area is titled 'Sản phẩm' (Products) and contains a table with 10 rows of product data. Each row includes a thumbnail image, ID, name, description, price, quantity, category, and an 'Edit' button. A red button at the top right says 'Thêm sản phẩm mới' (Add new product). At the bottom, there are navigation buttons for 'Previous' and 'Next'.

ID	TÊN	MÔ TẢ	GIÁ	SỐ LƯỢNG	PHÂN LOẠI	HÀNH ĐỘNG
1	Charm hoa Tulip trắng	Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay giết. Nguồn gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại xưởng gia công riêng, đảm bảo...	55.000đ	20	Charm cho nữ	Edit
2	Charm hoa Tulip tím	Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay giết. Nguồn gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại xưởng gia công riêng, đảm bả...	55.000đ	20	Charm cho nữ	Edit
3	Charm hoa Tulip hồng	Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay giết. Nguồn gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại xưởng gia công riêng, đảm bả...	55.000đ	20	Charm cho nữ	Edit
4	Charm hoa Tulip đỏ	Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay giết. Nguồn gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại xưởng gia công riêng, đảm bả...	55.000đ	20	Charm cho nữ	Edit
5	Charm hoa Sen	Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay giết. Nguồn gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại xưởng gia công riêng, đảm bả...	55.000đ	20	Charm cho nữ	Edit
6	Charm hoa Mini	Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay giết. Nguồn gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại xưởng gia công riêng, đảm bả...	55.000đ	20	Charm cho nữ	Edit
7	Charm hoa Ly	Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay giết. Nguồn gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại xưởng gia công riêng, đảm bả...	55.000đ	20	Charm cho nữ	Edit
8	Charm hoa Hướng Dương	Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay giết. Nguồn gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại xưởng gia công riêng, đảm bả...	55.000đ	20	Charm cho nữ	Edit
9	Charm logo BMW	Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay giết. Nguồn gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại xưởng gia công riêng, đảm bả...	59.000đ	20	Charm cho nam	Edit
10	Charm logo Bugatti	Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay giết. Nguồn gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại xưởng gia công riêng, đảm bả...	59.000đ	20	Charm cho nam	Edit

2.1.1 Các thành phần từ trái sang phải, từ trên xuống dưới:

- Logo Usbibracelet
- Icon Người dùng
- Icon Sản phẩm
- Icon Đơn hàng
- Icon Tin tức
- Thanh tìm kiếm
- NútThêm sản phẩm mới
- ID: Mã số sản phẩm
- TÊN: Tên sản phẩm
- MÔ TẢ: Mô tả chi tiết về sản phẩm
- GIÁ: Giá của sản phẩm
- SỐ LUỢNG: Số lượng sản phẩm hiện có
- PHÂN LOẠI: Phân loại các sản phẩm khác nhau
- HÀNH ĐỘNG: Có thể sửa lại thông tin sản phẩm
- Nút Edit

- Thanh xem sản phẩm ở các trang

2.1.2 Các tương tác

- Icon Người dùng

Mô tả: Khi người dùng ấn vào ô Người dùng, là đường dẫn tới Trang Quản lý người dùng.

Hiệu ứng: Ô Người dùng chuyển đỏ khi di chuột qua

Sự kiện: Khi người dùng ấn vào ô Người dùng, lập tức được điều hướng đến Trang Quản lý người dùng.



- Icon Sản phẩm

Mô tả: Khi người dùng ấn vào ô Sản phẩm, là đường dẫn tới Trang Quản lý Sản phẩm.

Hiệu ứng: Ô Sản phẩm chuyển đỏ khi di chuột qua

Sự kiện: Khi người dùng ấn vào ô Sản phẩm, lập tức được điều hướng đến Trang Quản lý Sản phẩm.

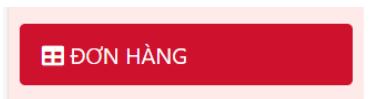


- Icon Đơn hàng

Mô tả: Khi người dùng ấn vào ô Đơn hàng, là đường dẫn tới Trang Quản lý đơn hàng.

Hiệu ứng: Ô Đơn hàng chuyển đỏ khi di chuột qua

Sự kiện: Khi người dùng ấn vào ô Người dùng, lập tức được điều hướng đến Trang Quản lý người dùng.



- Icon Tin tức

Mô tả: Khi người dùng ấn vào ô Tin tức, là đường dẫn tới Trang Quản lý Tin tức.

Hiệu ứng: Ô Tin tức chuyển đỏ khi di chuột qua

Sự kiện: Khi người dùng ấn vào ô Tin tức, lập tức được điều hướng đến Trang Quản lý Tin tức.



- Thanh tìm kiếm

Mô tả: Người dùng có thể tìm kiếm tên sản phẩm, nếu có sẽ hiển thị sản phẩm trên trang Quản lý sản phẩm.

Hiệu ứng: Không có

Sự kiện: Người dùng ấn vào ô tìm kiếm và nhập tên tên sản phẩm muốn tìm kiếm, màn hình trang Quản lý sản phẩm hiển thị tên sản phẩm nếu có.



- Nút Thêm sản phẩm mới

Mô tả: Người dùng ấn vào nút thêm sản phẩm mới, là đường dẫn link tới trang Chi tiết quản lý sản phẩm

Hiệu ứng: Không có

Sự kiện: Người dùng ấn vào nút thêm Sản phẩm mới, lập tức sẽ được điều hướng đến trang Chi tiết quản lý Sản phẩm

The screenshot shows a user interface for creating a new product. On the left, there's a sidebar with a logo 'USBIBRACELET' and four menu items: 'NGƯỜI DÙNG', 'SẢN PHẨM', 'ĐƠN HÀNG', and 'TIN TỨC'. The 'TIN TỨC' item is highlighted with a red box. The main content area is titled 'Create Product'. It contains several input fields: 'Product ID' (placeholder 'Enter ID'), 'Product Name' (placeholder 'Enter Product Name'), 'Description' (placeholder 'Enter Product Description'), 'Price' (placeholder 'Enter Product Price'), 'Quantity' (placeholder 'Enter Quantity'), 'Category' (a dropdown menu placeholder 'Select Category'), and 'Image Path' (placeholder 'Enter Image Path'). At the bottom is a large blue button labeled 'Create Product'.

Hình - Giao diện trang tạo Sản phẩm mới

- Nút Edit

Mô tả: Người dùng ấn vào nút Edit, là đường dẫn link tới trang Chi tiết quản lý Sản phẩm

Hiệu ứng: Không có

Sự kiện: Người dùng ấn vào nút Edit, lập tức sẽ được điều hướng đến trang Chi tiết quản lý Sản phẩm

The screenshot shows a user interface for updating a product. On the left, there's a sidebar with navigation links: NGƯỜI DÙNG, SẢN PHẨM, ĐƠN HÀNG, and TIN TỨC. The main area has a title 'Update Product'. It contains several input fields: 'Product ID' (1), 'Product Name' (Charm hoa Tulip trắng), 'Description' (Chất liệu: Thép không gỉ; Cam kết chất lượng: Charm không bị xin màu, đen hay gi sét.), 'Price' (55000.00), 'Quantity' (20), 'Category' (Charm cho nữ), and 'Image Path' (assets/images/Charmchonu/Charm hoa Tulip trắng 55k.jpg). At the bottom is a large blue 'Update Product' button.

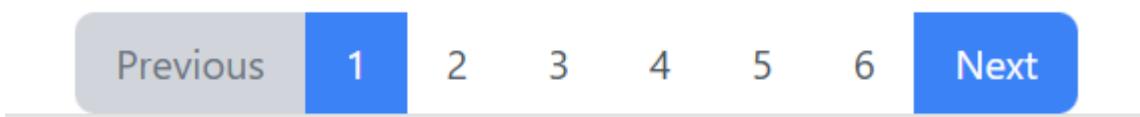
Hình - Giao diện Cập nhật lại Chi tiết sản phẩm

- Thanh xem sản phẩm ở các trang

Mô tả: Khi người dùng ấn vào nút Previous, Next hoặc số trang, chuyển đến trang trong Quản lý sản phẩm mà người dùng ấn chọn

Hiệu ứng: Không có

Sự kiện: Khi người dùng ấn vào nút Previous, Next hoặc số trang, lập tức chuyển đến trang sản phẩm trong Quản lý sản phẩm mà người dùng ấn chọn



2.2 Chức năng

Bước 1: Người dùng truy cập vào trang quản lý sản phẩm (product-management-detail.php)

Khi người quản lý vào trang **product-management-detail.php**, hệ thống sẽ thực hiện một số bước sau:

- **Lấy danh sách các danh mục sản phẩm:** Hệ thống lấy tất cả các danh mục từ bảng category để hiển thị trong dropdown cho người quản lý chọn khi tạo hoặc cập nhật sản phẩm.

```
$categories = [];
    // Lấy danh sách các danh mục từ bảng categories (giả sử có bảng riêng cho categories)
$stmt = $pdo->query("SELECT * FROM category");
$categories = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

- **Kiểm tra có productId trong URL hay không:** Nếu có productId trong URL (ví dụ: product-management.php?productId=5), hệ thống sẽ lấy thông tin chi tiết của sản phẩm từ bảng product, giá của sản phẩm từ bảng productprice, và hình ảnh của sản phẩm từ bảng image.

```
if (isset($_GET['productId'])) {
$product_id = $_GET['productId'];
// Truy vấn thông tin sản phẩm từ cơ sở dữ liệu
$stmt = $pdo->prepare("SELECT * FROM product WHERE id = :id");
$stmt->bindParam(":id", $product_id);
$stmt->execute();
$product = $stmt->fetch(PDO::FETCH_ASSOC);
// Lấy các hình ảnh của sản phẩm
$stmtImages = $pdo->prepare("SELECT path FROM image WHERE product_id = :productId");
$stmtImages->bindParam(':productId', $product_id, PDO::PARAM_INT);
$stmtImages->execute();
$images = $stmtImages->fetch(PDO::FETCH_ASSOC);
// Lấy giá sản phẩm
```

```
$stmtPrice = $pdo->prepare('SELECT * FROM productprice WHERE product_id = :productId');
$stmtPrice->bindParam(':productId', $product_id, PDO::PARAM_INT);
$stmtPrice->execute();
$price = $stmtPrice->fetch(PDO::FETCH_ASSOC);
```

- **Hiển thị thông tin sản phẩm lên form:** Nếu productId có trong URL, các trường thông tin như tên sản phẩm, mô tả, giá, hình ảnh, danh mục sẽ được điền sẵn vào form. Nếu không có productId, form sẽ rỗng, dành cho việc tạo mới sản phẩm.

Bước 2: Người quản lý điền thông tin sản phẩm vào form và gửi (POST đến product-management-detail.inc.php)

Khi người quản lý điền thông tin vào form và nhấn "Submit", các dữ liệu này sẽ được gửi đến file **product-management-detail.inc.php** qua phương thức POST. Trong form, thông tin được thu thập bao gồm:

- id: ID của sản phẩm (nếu có, khi cập nhật sản phẩm).
- name: Tên sản phẩm.
- description: Mô tả sản phẩm.
- price: Giá sản phẩm.
- quantity: Số lượng sản phẩm.
- category: ID danh mục sản phẩm.
- image: Đường dẫn ảnh sản phẩm (có thể rỗng hoặc có giá trị).

Bước 3: Xử lý trong product-management-detail.inc.php

Khi nhận được dữ liệu từ form gửi lên, file **product-management-detail.inc.php** sẽ thực hiện các bước xử lý sau:

- Kiểm tra danh mục sản phẩm: Trước tiên, hệ thống kiểm tra xem category_id (danh mục sản phẩm) có hợp lệ hay không, tức là có tồn tại trong bảng category không.

```

// Kiểm tra xem category_id có tồn tại trong bảng category không

$stmtCategory = $pdo->prepare("SELECT id FROM category WHERE id = :category_id");

$stmtCategory->bindParam(':category_id', $category_id);

$stmtCategory->execute();

$category = $stmtCategory->fetch(PDO::FETCH_ASSOC);

// Nếu category_id không tồn tại trong bảng category, dừng lại và thông báo lỗi

if (!$category) {

    echo "Danh mục không tồn tại trong cơ sở dữ liệu./";

    exit;

}

```

- Thêm mới hoặc cập nhật sản phẩm

Nếu sản phẩm đã tồn tại (có **id**): Thực hiện câu lệnh SQL để cập nhật thông tin sản phẩm. Chỉ các trường **name**, **description**, **category_id**, **quantity** sẽ được cập nhật.

```

// Nếu đã có id, tức là đang cập nhật sản phẩm

$stmt = $pdo->prepare("UPDATE product SET name = :name, description = :description, category_id = :category_id, quantity = :quantity WHERE id = :id");

$stmt->bindParam(':category_id', $category_id);
$stmt->bindParam(':name', $name);
$stmt->bindParam(':description', $description);
$stmt->bindParam(':quantity', $quantity);
$stmt->bindParam(':id', $id);

$stmt->execute();

```

Nếu sản phẩm chưa có id (tạo mới sản phẩm): Thực hiện câu lệnh SQL để thêm mới sản phẩm vào bảng product.

```
// Nếu chưa có id, tạo mới sản phẩm
$stmt = $pdo->prepare("INSERT INTO product (name, description, quantity,
category_id) VALUES (:name, :description, :quantity, :category_id)");
$stmt->bindParam(':category_id', $category_id);
$stmt->bindParam(':name', $name);
$stmt->bindParam(':description', $description);
$stmt->bindParam(':quantity', $quantity);
$stmt->execute();
```

- Cập nhật giá sản phẩm

Tiếp theo, hệ thống kiểm tra xem giá sản phẩm đã tồn tại trong bảng productprice chưa. Nếu có, nó sẽ **cập nhật giá mới**. Nếu chưa có, nó sẽ **thêm mới giá** vào bảng productprice.

```
// Cập nhật giá sản phẩm vào bảng productprice

$stmtPrice = $pdo->prepare("SELECT * FROM productprice WHERE
product_id = :product_id");

$stmtPrice->bindParam(':product_id', $id);

$stmtPrice->execute();

$existingPrice = $stmtPrice->fetch(PDO::FETCH_ASSOC);

if ($existingPrice) {

    // Nếu giá đã tồn tại, cập nhật giá mới

    $stmtUpdatePrice = $pdo->prepare("UPDATE productprice SET price = :price
WHERE product_id = :product_id");

    $stmtUpdatePrice->bindParam(':price', $price);

    $stmtUpdatePrice->bindParam(':product_id', $id);
```

```

$stmtUpdatePrice->execute();

} else {

    // Nếu giá chưa tồn tại, thêm mới giá

    $stmtInsertPrice = $pdo->prepare("INSERT INTO productprice (product_id,
price) VALUES (:product_id, :price)");

    $stmtInsertPrice->bindParam(':price', $price);

    $stmtInsertPrice->bindParam(':product_id', $id);

    $stmtInsertPrice->execute();

}

```

- Cập nhật hình ảnh sản phẩm

```

// Nếu có ảnh mới, xóa tất cả ảnh cũ của sản phẩm này

$stmtDeleteImages = $pdo->prepare("DELETE FROM image WHERE
product_id = :product_id");

$stmtDeleteImages->bindParam(':product_id', $id);

$stmtDeleteImages->execute();


// Thêm ảnh mới vào bảng image

$stmtInsertImage = $pdo->prepare("INSERT INTO image (product_id,
image_path) VALUES (:product_id, :image_path)");

$stmtInsertImage->bindParam(':product_id', $id);

$stmtInsertImage->bindParam(':image_path', $image);

$stmtInsertImage->execute();

```

- Chuyển hướng sau khi hoàn tất

Cuối cùng, sau khi xử lý xong, người quản lý sẽ được chuyển hướng về trang quản lý sản phẩm (product-management.php), và hệ thống sẽ thông báo kết quả (thành công hoặc lỗi).

```
// Sau khi hoàn thành, chuyển hướng về trang quản lý sản phẩm (hoặc trang nào đó)
```

```

header("Location: ./product-management.php");

$_SESSION['message'] = "Cập nhật sản phẩm thành công.";

exit;

```

3. Quản lý tin tức

3.1 Giao diện

The screenshot shows the Usbibracelet news management interface. On the left, there is a sidebar with icons for User, Product, Order, and News. The News icon is selected. The main area has a search bar at the top. Below it, a red button says "Thêm tin tức mới". The news list contains five items:

- Quà tặng cá nhân cho bạn bè: Ý tưởng tạo ra một món đồ trang sức độc đáo (December 14, 2024)
- Mua gì tặng bố mẹ vào dịp Giáng sinh (December 14, 2024)
- Quà tặng tốt nghiệp (December 14, 2024)
- 20 tuổi: một món quà sinh nhật độc đáo và mang tính cá nhân (December 14, 2024)
- Ngày của Mẹ: những câu nói hay dành tặng mẹ (December 14, 2024)

Hình - Giao diện trang Quản lý tin tức

The screenshot shows the Usbibracelet news management interface with a new news item creation form. The sidebar on the left is identical to the previous screenshot. The main area has a search bar at the top. A large text input field is labeled "Tiêu đề" with placeholder text "Nhập tiêu đề để tin tức". Below it is a rich text editor toolbar with various formatting options like bold, italic, underline, etc. At the bottom of the form is a red button labeled "Thêm mới tin tức".

3.1.1 Các thành phần từ trái sang phải, từ trên xuống dưới:

- Logo Usbibracelet
- Icon Người dùng
- Icon Sản phẩm

- Icon Đơn hàng
- Icon Tin tức
- Thanh tìm kiếm
- Nút thêm tin tức mới
- Tên tin tức
- Ngày tháng năm đăng tin tức
- Nút chỉnh sửa tin tức

3.1.2 Các tương tác

- Icon Người dùng

Mô tả: Khi người dùng ấn vào ô Người dùng, là đường dẫn tới Trang Quản lý người dùng.

Hiệu ứng: Ô Người dùng chuyển đỏ khi di chuột qua

Sự kiện: Khi người dùng ấn vào ô Người dùng, lập tức được điều hướng đến Trang Quản lý người dùng.



- Icon Sản phẩm

Mô tả: Khi người dùng ấn vào ô Sản phẩm, là đường dẫn tới Trang Quản lý Sản phẩm.

Hiệu ứng: Ô Sản phẩm chuyển đỏ khi di chuột qua

Sự kiện: Khi người dùng ấn vào ô Sản phẩm, lập tức được điều hướng đến Trang Quản lý Sản phẩm.



- Icon Đơn hàng

Mô tả: Khi người dùng ấn vào ô Đơn hàng, là đường dẫn tới Trang Quản lý đơn hàng.

Hiệu ứng: Ô Đơn hàng chuyển đỏ khi di chuột qua

Sự kiện: Khi người dùng ấn vào ô Người dùng, lập tức được điều hướng đến Trang Quản lý người dùng.

 ĐƠN HÀNG

- Icon Tin tức

Mô tả: Khi người dùng ấn vào ô Tin tức, là đường dẫn tới Trang Quản lý Tin tức.

Hiệu ứng: Ô Tin tức chuyển đỏ khi di chuột qua

Sự kiện: Khi người dùng ấn vào ô Tin tức, lập tức được điều hướng đến Trang Quản lý Tin tức.

 TIN TỨC

- Thanh tìm kiếm

Mô tả: Người dùng có thể tìm kiếm tên blog, nếu có sẽ hiển thị blog trên trang Quản lý Blog.

Hiệu ứng: Không có

Sự kiện: Người dùng ấn vào ô tìm kiếm và nhập tên tên Blog muốn tìm kiếm, màn hình trang Quản lý Blog hiển thị tên Blog nếu có.

Tìm kiếm

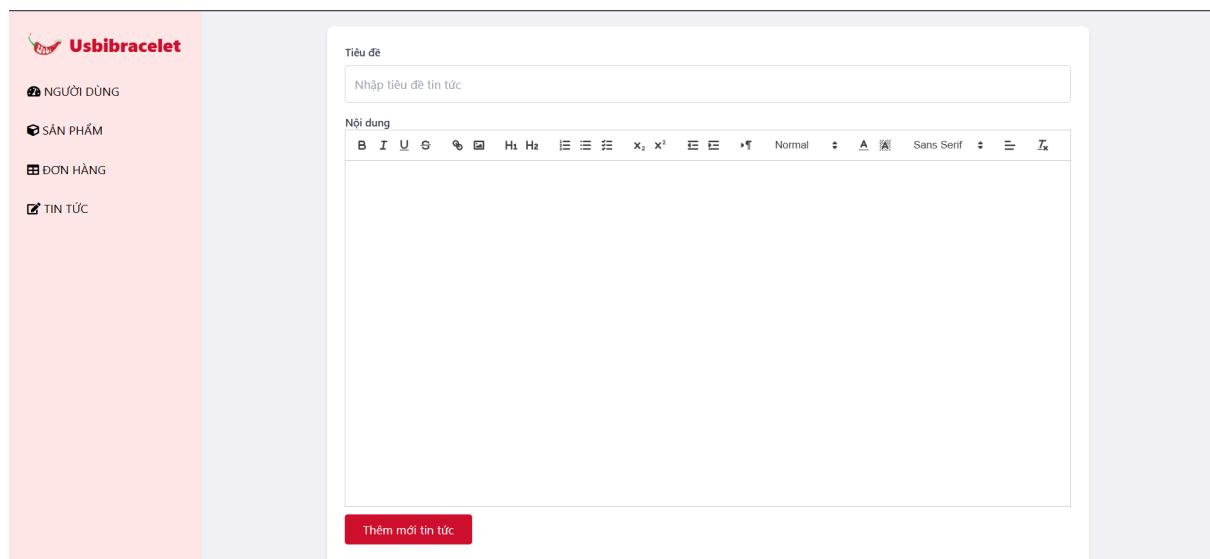
- Nút Thêm tin tức mới

Mô tả: Người dùng ấn vào nút thêm tin tức mới, là đường dẫn link tới trang Chi tiết quản lý tin tức

Hiệu ứng: Không có

Sự kiện: Người dùng ấn vào nút thêm tin tức mới, lập tức sẽ được điều hướng đến trang Chi tiết quản lý Tin tức

Thêm tin tức mới



Hình - Giao diện trang Chi tiết quản lý tin tức

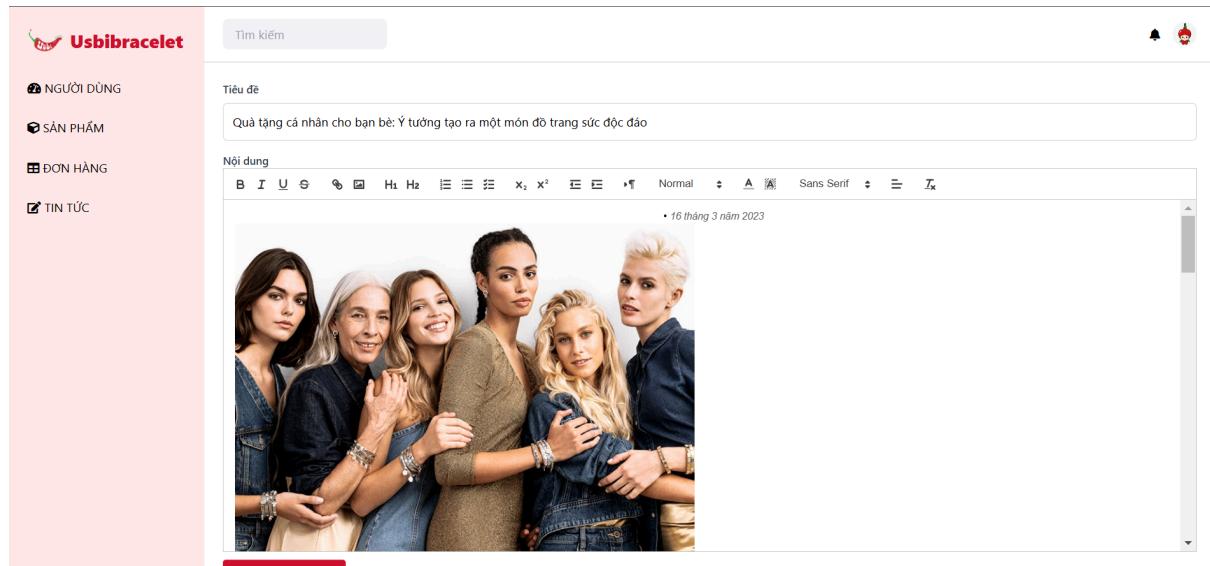
- Nút Chính sửa

Mô tả: Người dùng ấn vào nút Chính sửa, là đường dẫn link tới trang Chi tiết quản lý tin tức

Hiệu ứng: Không có

Sự kiện: Người dùng ấn vào nút Chính sửa, lập tức sẽ được điều hướng đến trang Chi tiết quản lý Tin tức

Chỉnh sửa



Hình - Giao diện trang Chi tiết Quản lý tin tức

3.2 Chức năng

Quản lý chi tiết từng bài viết:

Khởi tạo Session và Kết nối Cơ sở Dữ liệu:

```
session_start();  
require_once "db.inc.php";
```

- session_start(): Khởi tạo session để có thể lưu trữ và truy xuất các dữ liệu phiên (session), ví dụ như ID người dùng.
- require_once "db.inc.php": Bao gồm tệp kết nối cơ sở dữ liệu (PDO) để thực hiện các truy vấn.

Kiểm tra Phương Thức Request:

```
if ($_SERVER['REQUEST_METHOD'] === 'POST')
```

- \$_SERVER['REQUEST_METHOD']: Kiểm tra xem phương thức HTTP của yêu cầu hiện tại có phải là POST hay không (được gửi từ form).

Kiểm Tra và Lấy Dữ Liệu từ POST:

```
if (isset($_POST['title']) && isset($_POST['content'])) {  
    $title = $_POST['title'];  
    $content = $_POST['content'];  
    $accountId = $_SESSION['user_id'] ?? 0;  
    $blogId = $_POST['blogId'] ?? null;
```

- Kiểm tra sự tồn tại của các trường title và content trong dữ liệu POST: Nếu chúng không có, mã sẽ không tiếp tục xử lý.
- Lấy dữ liệu từ POST:
 - + \$_POST['title']: Tiêu đề bài viết.
 - + \$_POST['content']: Nội dung bài viết.
 - + \$_SESSION['user_id']: ID người dùng từ session (người đăng nhập). Nếu không có, mặc định là 0.
 - + \$_POST['blogId']: ID bài viết cần chỉnh sửa, nếu có (nếu không có thì sẽ là null).

Cập nhật hoặc Tạo Mới Bài Viết:

- **Cập nhật bài viết:**

- Nếu có blogId (được cung cấp khi chỉnh sửa bài viết), thực hiện truy vấn UPDATE để cập nhật tiêu đề và nội dung bài viết cũ.

```
$stmt = $pdo->prepare("UPDATE blog SET title = :title, content = :content, timestamp = NOW() WHERE id = :blogId AND account_id = :account_id");
$stmt->bindParam(':blogId', $blogId, PDO::PARAM_INT);
```

- UPDATE: Cập nhật các thông tin bài viết với ID bài viết (blogId) và ID tài khoản (accountId).
- Dùng NOW() để cập nhật thời gian hiện tại.

- Tạo mới bài viết:

- Nếu không có blogId, tức là người dùng muốn tạo bài viết mới, thực hiện truy vấn INSERT để thêm một bài viết mới vào cơ sở dữ liệu.

```
$stmt = $pdo->prepare("INSERT INTO blog (title, content, timestamp, account_id) VALUES (:title, :content, NOW(), :account_id)");
```

Gắn Các Tham Số và Thực Thi Truy Vấn:

```
$stmt->bindParam(':title', $title);
$stmt->bindParam(':content', $content);
$stmt->bindParam(':account_id', $accountId);
$stmt->execute();
```

- bindParam: Gắn giá trị vào các tham số trong truy vấn SQL để tránh SQL Injection.
- execute(): Thực thi câu lệnh SQL đã chuẩn bị.

Phản Hồi Thành Công hoặc Lỗi

```
if ($blogId) {
    echo "Blog updated successfully!";
} else {
    echo "Blog created successfully!";
}
```

- Nếu có blogId, trả về thông báo "Blog updated successfully!" (Cập nhật thành công).
- Nếu không có blogId, trả về thông báo "Blog created successfully!" (Tạo mới thành công).

Xử Lý Lỗi

```
catch (PDOException $e) {  
    // Trả về thông báo lỗi nếu gặp sự cố  
    echo "Error: " . $e->getMessage();
```

- Nếu xảy ra lỗi trong quá trình truy vấn cơ sở dữ liệu (ví dụ: kết nối bị lỗi, câu lệnh SQL không hợp lệ), mã sẽ bắt lỗi và hiển thị thông báo lỗi.

Kiểm Tra Dữ Liệu POST:

```
else {  
    echo "Title and content are required!";  
}
```

- Nếu thiếu title hoặc content trong dữ liệu POST, mã sẽ thông báo yêu cầu người dùng nhập đầy đủ.

Kiểm Tra Phương Thức Request:

```
else {  
    echo "Invalid request method!";
```

- Nếu phương thức HTTP không phải là POST (chẳng hạn như người dùng truy cập trực tiếp URL mà không gửi form), mã sẽ thông báo yêu cầu gửi đúng phương thức.

4. Quản lý đơn hàng

4.1 Giao diện

The screenshot shows the Usbibracelet website's order management page. On the left is a sidebar with navigation links: NGƯỜI DÙNG, SẢN PHẨM, ĐƠN HÀNG (selected), and TIN TỨC. A search bar labeled "Tim kiếm" is at the top. The main area is titled "Đơn hàng" and contains a table with 11 rows of order data. The columns are: ID, Tên, Số điện thoại, Địa chỉ, Vận chuyển, Thanh toán, Tổng giá trị, and Tình trạng. Each row includes a "Xem" button. At the bottom of the table are two small buttons: a red one labeled "1" and a grey one labeled "2".

ID	Tên	Số điện thoại	Địa chỉ	Vận chuyển	Thanh toán	Tổng giá trị	Tình trạng
2	Phạm Thị Mỹ Tâm	0336189732	Thôn 1, Cư Ebur, Tp Buôn Ma Thuột	doityourself	cod	110.000.000đ	pending
3	Phạm Thị Mỹ Tâm	0336189732	Thôn 1, Cư Ebur, Tp Buôn Ma Thuột	doityourself	cod	110.000.000đ	pending
4	Phạm Thị Mỹ Tâm	0336189732	Thôn 1, Cư Ebur, Tp Buôn Ma Thuột	doityourself	cod	110.000.000đ	pending
5	Phạm Thị Mỹ Tâm	0336189732	Thôn 1, Cư Ebur, Tp Buôn Ma Thuột	doityourself	cod	110.000.000đ	pending
6	Phạm Thị Mỹ Tâm	0336189732	Thôn 1, Cư Ebur, Tp Buôn Ma Thuột	bee	momo	130.025.000đ	successed
7	Phạm Thị Mỹ Tâm	0336189732	Thôn 1, Cư Ebur, Tp Buôn Ma Thuột	doityourself	cod	30.000.000đ	successed
8	Phạm Thị Mỹ Tâm	0336189732	Thôn 1, Cư Ebur, Tp Buôn Ma Thuột	bee		52.025.000đ	successed
9	Phạm Thị Mỹ Tâm	0336189732	Thôn 1, Cư Ebur, Tp Buôn Ma Thuột	bee	cod	40.025.000đ	pending
10	Phạm Thị Mỹ Tâm	0336189732	Thôn 1, Cư Ebur, Tp Buôn Ma Thuột	bee	cod	70.025.000đ	successed
11	Phạm Thị Mỹ Tâm	522 Nguyễn Trí Phương		doityourself	cod	220.000đ	pending

Hình - Giao diện trang “Quản lý đơn hàng”

Giao diện của trang quản lý đơn hàng gồm các thành phần chính sau:

Thanh điều hướng (Sidebar): Bên trái màn hình là menu điều hướng, bao gồm các mục: **Người dùng, Sản phẩm, Đơn hàng, Tin tức.**

Tìm kiếm: Phía trên bên phải có một ô tìm kiếm cho phép người dùng tìm các đơn hàng theo tên hoặc thông tin khác.

Danh sách đơn hàng: Bảng hiển thị các đơn hàng với các cột thông tin gồm:

- **ID:** Mã số đơn hàng.
- **Tên:** Tên khách hàng (Phạm Thị Mỹ Tâm trong các đơn hàng trên).
- **Số điện thoại:** Số điện thoại của khách hàng.
- **Địa chỉ:** Địa chỉ giao hàng của khách hàng.
- **Vận chuyển:** Hình thức vận chuyển (ví dụ: doityourself, bee).
- **Thanh toán:** Phương thức thanh toán (ví dụ: COD, momo).
- **Tổng giá trị:** Tổng giá trị của đơn hàng.

- **Tình trạng:** Trạng thái của đơn hàng (ví dụ: pending, succeeded), có các nút "Xem" để xem chi tiết.

The screenshot shows a user interface for managing orders. On the left, there's a sidebar with navigation links: NGƯỜI DÙNG, SẢN PHẨM, ĐƠN HÀNG, and TIN TỨC. The main content area has a header 'Chi tiết đơn hàng'. Below it, there's a summary table with columns for ID, Tên khách hàng, Số điện thoại, Địa chỉ, Hình thức vận chuyển, Hình thức thanh toán, Tổng giá trị, and Tình trạng. The summary table shows the following data:

ID đơn hàng: 2	Tên khách hàng: Phạm Thị Mỹ Tâm	Số điện thoại: 0336189732	Địa chỉ: thôn 1, Cư Ebur, TP Buôn Ma Thuột	Hình thức vận chuyển: doityourself	Hình thức thanh toán: cod	Tổng giá trị: 110.000.000đ	Tình trạng: pending
----------------	---------------------------------	---------------------------	--	------------------------------------	---------------------------	----------------------------	---------------------

A separate table titled 'Products' is shown below, with columns for Product Name, Quantity, Price, and Subtotal. The table is currently empty.

Hình - Giao diện trang “Chi tiết quản lý đơn hàng”

Giao diện trang “Chi tiết quản lý đơn hàng” cho phép người dùng theo dõi chi tiết các đơn hàng, bao gồm thông tin cá nhân của khách hàng, phương thức thanh toán và vận chuyển, cùng với danh sách các sản phẩm và tổng giá trị đơn hàng.

Phần Chi tiết đơn hàng: Hiển thị thông tin chi tiết về đơn hàng, bao gồm.

- **ID đơn hàng** (ví dụ: 2)
- **Tên khách hàng** (ví dụ: Phạm Thị Mỹ Tâm)
- **Số điện thoại** (ví dụ: 0336189732)
- **Địa chỉ** (ví dụ: thôn 1, Cư Ebur, TP Buôn Ma Thuột)
- **Hình thức vận chuyển** (ví dụ: doityourself)
- **Hình thức thanh toán** (ví dụ: COD - trả tiền khi nhận hàng)
- **Tổng giá trị** (ví dụ: 110.000.000đ)
- **Tình trạng đơn hàng** (ví dụ: pending - đang chờ xử lý)

Danh sách sản phẩm trong đơn hàng: Một bảng chứa các thông tin về các sản phẩm trong đơn hàng, bao gồm.

- **Tên sản phẩm**
- **Số lượng**
- **Giá**
- **Tổng giá trị của sản phẩm**

4.2 Chức năng

Bước 1: Hiển thị danh sách đơn hàng trên trang order-management.php

Trong order-management.php, chúng ta sẽ lấy danh sách các đơn hàng từ cơ sở dữ liệu và hiển thị chúng trên bảng.

Code phần hiển thị danh sách đơn hàng (trong order-management.php):

```
<?php
session_start();
require_once "include/db.inc.php";

// Get current page for pagination (default to 1 if not set)
$page = isset($_GET['page']) ? (int)$_GET['page'] : 1;
$perPage = 10; // Number of orders per page
$offset = ($page - 1) * $perPage;

$sql = "
SELECT
    o.id,
    a.firstname,
    a.lastname,
    o.phonenumber,
    o.address,
    o.shippingMethod,
    o.paymentMethod,
    o.total,
    o.status
FROM `order` AS o
```

```
JOIN `account` AS a ON o.account_id = a.id  
ORDER BY o.id ASC  
LIMIT :offset, :perPage  
";
```

- Truy vấn SQL này sẽ lấy danh sách đơn hàng từ cơ sở dữ liệu, bao gồm thông tin khách hàng (tên, số điện thoại, địa chỉ, v.v.) và các chi tiết đơn hàng như phương thức giao hàng, thanh toán, tổng tiền và trạng thái.
- Các đơn hàng được phân trang với số lượng mỗi trang là 10 (\$perPage).
- Dữ liệu từ cơ sở dữ liệu được đưa vào mảng \$orders.

Hiển thị danh sách đơn hàng:

```
<table class="min-w-full bg-white">  
  <thead class="bg-gray-50">  
    <tr>  
      <th class="py-2 px-4 border-b">ID</th>  
      <th class="py-2 px-4 border-b">Name</th>  
      <th class="py-2 px-4 border-b">Phone</th>  
      <th class="py-2 px-4 border-b">Address</th>  
      <th class="py-2 px-4 border-b">Shipping</th>  
      <th class="py-2 px-4 border-b">Payment</th>  
      <th class="py-2 px-4 border-b">Total</th>  
      <th class="py-2 px-4 border-b">Status</th>  
      <th class="py-2 px-4 border-b"></th>  
    </tr>  
  </thead>
```

```
<tbody>

<?php if ($orders): ?>

<?php foreach ($orders as $order): ?>

<tr class="hover:bg-gray-50">

    <td class="py-2 px-4 border-b"><?= htmlspecialchars($order['id']) ?></td>

        <td class="py-2 px-4 border-b"><?= htmlspecialchars($order['firstname']) ?> <?= htmlspecialchars($order['lastname']) ?></td>

        <td class="py-2 px-4 border-b"><?= htmlspecialchars($order['phonenumber']) ?></td>

        <td class="py-2 px-4 border-b"><?= htmlspecialchars($order['address']) ?></td>

        <td class="py-2 px-4 border-b"><?= htmlspecialchars($order['shippingMethod']) ?></td>

        <td class="py-2 px-4 border-b"><?= htmlspecialchars($order['paymentMethod']) ?></td>

        <td class="py-2 px-4 border-b"><?= number_format($order['total'], 0, ',', '.') ?>&lt;/td>

        <td class="py-2 px-4 border-b"><?= htmlspecialchars($order['status']) ?></td>

        <td class="py-2 px-4 border-b">

            <a class="text-blue-500 underline" href="order-detail.php?orderId=<?= $order['id'] ?>">View</a>

        </td>
```

```

        </tr>

        <?php endforeach; ?>

        <?php else: ?>

        <tr>

            <td colspan="8" class="py-4 text-center text-gray-500">No orders
found.</td>

        </tr>

        <?php endif; ?>

    </tbody>

</table>

```

- Đây là phần HTML để hiển thị danh sách đơn hàng trong bảng. Các thông tin như tên, số điện thoại, địa chỉ, phương thức giao hàng, thanh toán, tổng tiền và trạng thái của mỗi đơn hàng được hiển thị.
- Mỗi dòng đơn hàng có một liên kết "View" ([View](order-detail.php?orderId=<?= $order['id'] ?>)) mà khi người dùng nhấp vào sẽ dẫn đến trang order-detail.php và truyền tham số orderId (ID của đơn hàng) qua URL.

Bước 2: Chuyển sang trang order-detail.php

Khi người dùng nhấp vào liên kết "View", họ sẽ được chuyển đến trang order-detail.php, trong đó tham số orderId sẽ được truyền qua URL. Ví dụ: order-detail.php?orderId=5.

Lấy orderId từ URL trong order-detail.php:

```
$orderId = isset($_GET['orderId']) ? (int)$_GET['orderId'] : 0;
```

Mã này lấy tham số orderId từ URL và ép kiểu nó thành kiểu int. Nếu không có tham số orderId, giá trị mặc định là 0.

Truy vấn thông tin chi tiết đơn hàng từ cơ sở dữ liệu:

Truy vấn thông tin đơn hàng từ bảng order và bảng account để lấy các thông tin chi tiết như tên khách hàng, địa chỉ, phương thức thanh toán, v.v.

```
$sqlOrder = "
```

```
SELECT
```

```
    o.id AS order_id,
```

```
    a.firstname,
```

```
    a.lastname,
```

```
    o.phonenumber,
```

```
    o.address,
```

```
    o.shippingMethod,
```

```
    o.paymentMethod,
```

```
    o.total,
```

```
    o.status
```

```
FROM `order` AS o
```

```
JOIN `account` AS a ON o.account_id = a.id
```

```
WHERE o.id = :orderId
```

```
";
```

Truy vấn này lấy thông tin chi tiết về các sản phẩm trong đơn hàng dựa trên orderId từ bảng orderdetail và bảng product. Kết quả sẽ trả về tên sản phẩm, số lượng và giá của từng sản phẩm trong đơn hàng.

Bước 3: Hiển thị chi tiết đơn hàng

Sau khi truy vấn dữ liệu từ cơ sở dữ liệu, chúng ta hiển thị thông tin chi tiết của đơn hàng, bao gồm thông tin khách hàng, tổng tiền, phương thức giao hàng, thanh toán và trạng thái đơn hàng. Ngoài ra, chúng ta cũng hiển thị bảng các sản phẩm trong đơn hàng.

Hiển thị thông tin đơn hàng và sản phẩm trong order-detail.php:

```
h2 class="text-xl font-semibold mb-4">Order Details
```

```
<p><strong>Order ID:</strong> <?= htmlspecialchars($orderInfo['order_id']) ?></p>
```

```
<p><strong>Customer Name:</strong> <?= htmlspecialchars($orderInfo['firstname'] . ' ' . $orderInfo['lastname']) ?></p>
```

```
<p><strong>Phone:</strong> <?= htmlspecialchars($orderInfo['phonenumber']) ?></p>
```

```
<p><strong>Address:</strong> <?= htmlspecialchars($orderInfo['address']) ?></p>
```

```
<p><strong>Shipping Method:</strong> <?= htmlspecialchars($orderInfo['shippingMethod']) ?></p>
```

```
<p><strong>Payment Method:</strong> <?= htmlspecialchars($orderInfo['paymentMethod']) ?></p>
```

```
<p><strong>Total:</strong> <?= number_format($orderInfo['total'], 0, ',', '.') ?>&dlt;/p>
```

```
<p><strong>Status:</strong> <?= htmlspecialchars($orderInfo['status']) ?></p>
```

```
<!-- Bảng chi tiết sản phẩm -->
```

```
<h3 class="text-lg font-semibold mt-6">Products</h3>

<table class="min-w-full bg-white border mt-4">

<thead class="bg-gray-50">

<tr>

<th class="py-2 px-4 border">Product Name</th>

<th class="py-2 px-4 border">Quantity</th>

<th class="py-2 px-4 border">Price</th>

<th class="py-2 px-4 border">Subtotal</th>

</tr>

</thead>

<tbody>

<?php foreach ($orderDetails as $detail): ?>

<tr>

<td class="py-2 px-4 border"><?= htmlspecialchars($detail['name']) ?></td>

<td class="py-2 px-4 border"><?= htmlspecialchars($detail['quantity']) ?></td>

<td class="py-2 px-4 border"><?= number_format($detail['price'], 0, ',', ',') ?>&dlt;/td>

<td class="py-2 px-4 border"><?= number_format($detail['quantity'] * $detail['price'], 0, ',', ',') ?>&dlt;/td>

</tr>

<?php endforeach; ?>
```

```
</tbody>
```

```
</table>
```

CHƯƠNG C: KẾT NỐI DATABASE

1. Tạo cơ sở dữ liệu



Nhấn vào create là hoàn thành bước tạo cơ sở dữ liệu

2. Tạo các bảng

Thực hiện tạo bảng trong mysql, các bảng được tạo bao gồm:

Bảng account:

```
30 • Ⓜ CREATE TABLE `account` (
31     `id` int(11) NOT NULL,
32     `email` varchar(255) NOT NULL,
33     `password` varchar(255) NOT NULL,
34     `firstname` varchar(100) NOT NULL,
35     `lastname` varchar(100) NOT NULL,
36     `role` enum('user','admin') NOT NULL
37 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Bảng blog:

```
52 • Ⓜ CREATE TABLE `blog` (
53     `id` int(11) NOT NULL,
54     `title` varchar(255) NOT NULL,
55     `timestamp` datetime NOT NULL,
56     `content` text NOT NULL,
57     `account_id` int(11) NOT NULL
58 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Bảng category:

```
66 • Ⓜ CREATE TABLE `category` (
67     `id` int(11) NOT NULL,
68     `name` varchar(255) NOT NULL,
69     `description` text DEFAULT NULL
70 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Bảng image:

```
91 • CREATE TABLE `image` (
92     `id` int(11) NOT NULL,
93     `path` varchar(255) NOT NULL,
94     `product_id` int(11) DEFAULT NULL,
95     `blog_id` int(11) DEFAULT NULL
96 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Bảng order:

```
170 • CREATE TABLE `order` (
171     `id` int(11) NOT NULL,
172     `account_id` int(11) NOT NULL,
173     `timestamp` datetime NOT NULL DEFAULT current_timestamp(),
174     `status` enum('pending','canceled','successed') NOT NULL DEFAULT 'pending',
175     `phonenumber` varchar(10) NOT NULL,
176     `provinceId` int(11) NOT NULL,
177     `districtId` int(11) NOT NULL,
178     `wardId` int(11) NOT NULL,
179     `address` text NOT NULL,
180     `shippingMethod` enum('doityourself','grab','bee') NOT NULL,
181     `paymentMethod` enum('cod','banking','momo') NOT NULL,
182     `total` int(11) NOT NULL
183 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Bảng orderdetail:

```
206 • CREATE TABLE `orderdetail` (
207     `id` int(11) NOT NULL,
208     `product_id` int(11) NOT NULL,
209     `quantity` int(11) NOT NULL,
210     `price` decimal(10,2) NOT NULL,
211     `order_id` int(11) NOT NULL
212 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Bảng product:

```
233 • CREATE TABLE `product` (
234     `id` int(11) NOT NULL,
235     `name` varchar(255) NOT NULL,
236     `description` text DEFAULT NULL,
237     `quantity` int(11) NOT NULL,
238     `category_id` int(11) NOT NULL
239 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Bảng productprice:

```
313 • CREATE TABLE `productprice` (
314     `id` int(11) NOT NULL,
315     `price` decimal(10,2) NOT NULL,
316     `starting_timestamp` datetime NOT NULL,
317     `ending_timestamp` datetime DEFAULT NULL,
318     `product_id` int(11) NOT NULL
319 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

3. Thêm dữ liệu vào bảng

Thêm dữ liệu cho bảng category:

```
76 • INSERT INTO `category` (`id`, `name`, `description`) VALUES
77     (1, 'Charm cho nữ', 'Bao gồm charm các loại hoa'),
78     (2, 'Charm cho nam', 'Bao gồm charm trơn và charm logo'),
79     (3, 'Charm lủng lẳng', 'Bao gồm các charm có sticker vui nhộn'),
80     (4, 'Charm chữ số', 'Bao gồm các charm chữ và charm số'),
81     (5, 'Sample mix sắn', 'Bao gồm các vòng tay có charm được mix sắn'),
82     (6, 'Charm dài', 'Bao gồm các charm có độ dài lớn hơn các charm thông thường'),
83     (7, 'Charm đính đá', 'Bao gồm các charm được đính đá lên sắn');
```

Thêm dữ liệu cho bảng image:

```
102 • INSERT INTO `image` (`id`, `path`, `product_id`, `blog_id`) VALUES
103     (1, 'assets/images/Charmchonu/Charm hoa Tulip trắng 55k.jpg', 1, NULL),
104     (2, 'assets/images/Charmchonu/Charm hoa Tulip tím 55k.jpg', 2, NULL),
105     (3, 'assets/images/Charmchonu/Charm hoa Tulip hồng 55k.jpg', 3, NULL),
106     (4, 'assets/images/Charmchonu/Charm hoa Tulip đỏ 55k.jpg', 4, NULL),
107     (5, 'assets/images/Charmchonu/Charm hoa Sen 55k.png', 5, NULL),
108     (6, 'assets/images/Charmchonu/Charm hoa Mini trắng 55k.jpg', 6, NULL),
109     (7, 'assets/images/Charmchonu/Charm hoa Ly 55k.png', 7, NULL),
110     (8, 'assets/images/Charmchonu/Charm hoa Hướng Dương 55k.jpg', 8, NULL),
111     (9, 'assets/images/Charmchonam/Charm logo BMW 59k.jpg', 9, NULL),
112     (10, 'assets/images/Charmchonam/Charm logo Bugati 59k.png', 10, NULL),
113     (11, 'assets/images/Charmchonam/Charm logo Ferrari 59k.jpg', 11, NULL),
114     (12, 'assets/images/Charmchonam/Charm logo Ford 59k.png', 12, NULL),
115     (13, 'assets/images/Charmchonam/Charm logo Mercedes 59k.jpg', 13, NULL),
116     (14, 'assets/images/Charmchonam/Charm logo Porsche 59k.jpg', 14, NULL),
117     (15, 'assets/images/Charmchonam/Charm logo Rolls Royce 59k.png', 15, NULL),
118     (16, 'assets/images/Charmchonam/Charm logo Volvo 59k.png', 16, NULL),
119     (17, 'assets/images/Charmlunglang/Charm cây thông Noel 55k.jpg', 17, NULL),
120     (18, 'assets/images/Charmlunglang/Charm chuông Noel 55k.jpg', 18, NULL),
121     (19, 'assets/images/Charmlunglang/Charm mũ len Noel 55k.jpg', 19, NULL),
122     (20, 'assets/images/Charmlunglang/Charm người tuyêt Noel 59k.jpg', 20, NULL),
```

Thêm dữ liệu cho bảng product:

```

245 • INSERT INTO `product` (`id`, `name`, `description`, `quantity`, `category_id`) VALUES
246     (1, 'Charm hoa Tulip trắng', 'Chất liệu: Thép không gỉ\nCam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét.\r\nNguyên gốc rõ ràng: Charm được sản xuất theo quy trình c
247     (2, 'Charm hoa Tulip tím', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn t
248     (3, 'Charm hoa Tulip hồng', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn t
249     (4, 'Charm hoa Tulip đỏ', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại
250     (5, 'Charm hoa Sen', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại xưở
251     (6, 'Charm hoa Mini', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại xuố
252     (7, 'Charm hoa Ly', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại xưởng
253     (8, 'Charm hoa Hướng Dương', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn
254     (9, 'Charm logo BMW', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại xuố
255     (10, 'Charm logo Bugatti', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn t
256     (11, 'Charm logo Ferrari', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn t
257     (12, 'Charm logo Ford', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại x
258     (13, 'Charm logo Mercedes', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn t
259     (14, 'Charm logo Porsche', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn t
260     (15, 'Charm logo Rolls Royce', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn
261     (16, 'Charm logo Volvo', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại
262     (17, 'Charm Cây thông Noel', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn
263     (18, 'Charm Chuông Noel', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại
264     (19, 'Charm Mũ len Noel', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn tại
265     (20, 'Charm Người tuyêt Noel', 'Chất liệu: Thép không gỉ. Cam kết chất lượng: Charm không bị xỉn màu, đèn hay gi sét. Nguyên gốc rõ ràng: Charm được sản xuất theo quy trình chuẩn

```

Thêm dữ liệu cho bảng productprice:

```

325 • INSERT INTO `productprice` (`id`, `price`, `starting_timestamp`, `ending_timestamp`, `product_id`) VALUES
326     (1, 55000.00, '2024-12-11 22:00:00', NULL, 1),
327     (2, 55000.00, '2024-12-11 22:00:00', NULL, 2),
328     (3, 55000.00, '2024-12-11 22:00:00', NULL, 3),
329     (4, 55000.00, '2024-12-11 22:00:00', NULL, 4),
330     (5, 55000.00, '2024-12-11 22:00:00', NULL, 5),
331     (6, 55000.00, '2024-12-11 22:00:00', NULL, 6),
332     (7, 55000.00, '2024-12-11 22:00:00', NULL, 7),
333     (8, 55000.00, '2024-12-11 22:00:00', NULL, 8),
334     (9, 59000.00, '2024-12-11 22:00:00', NULL, 9),
335     (10, 59000.00, '2024-12-11 22:00:00', NULL, 10),
336     (11, 59000.00, '2024-12-11 22:00:00', NULL, 11),
337     (12, 59000.00, '2024-12-11 22:00:00', NULL, 12),
338     (13, 59000.00, '2024-12-11 22:00:00', NULL, 13),
339     (14, 59000.00, '2024-12-11 22:00:00', NULL, 14),
340     (15, 59000.00, '2024-12-11 22:00:00', NULL, 15),
341     (16, 59000.00, '2024-12-11 22:00:00', NULL, 16),
342     (17, 55000.00, '2024-12-11 22:00:00', NULL, 17),
343     (18, 55000.00, '2024-12-11 22:00:00', NULL, 18),
344     (19, 55000.00, '2024-12-11 22:00:00', NULL, 19),
345     (20, 59000.00, '2024-12-11 22:00:00', NULL, 20),

```

4. Tạo các mối quan hệ giữa các bảng bằng khóa ngoại

Ràng buộc khóa chính:

```

394 • ALTER TABLE `account`
395         ADD PRIMARY KEY (`id`);
396
397         --
398         -- Indexes for table `blog`
399         --
400 • ALTER TABLE `blog`
401         ADD PRIMARY KEY (`id`),
402         ADD KEY `account_id` (`account_id`);

```

```
407 ●    ALTER TABLE `category`
408         ADD PRIMARY KEY (`id`);
409
410     --
411     -- Indexes for table `image`
412     --
413 ●    ALTER TABLE `image`
414         ADD PRIMARY KEY (`id`),
415         ADD KEY `product_id` (`product_id`),
416         ADD KEY `blog_id` (`blog_id`);
417
418     --
419     -- Indexes for table `order`
420     --
421 ●    ALTER TABLE `order`
422         ADD PRIMARY KEY (`id`),
423         ADD KEY `account_id` (`account_id`);
424
425 ●    ALTER TABLE `orderdetail`
426         ADD PRIMARY KEY (`id`),
427         ADD KEY `product_id` (`product_id`),
428         ADD KEY `order_id` (`order_id`);
429
430     --
431     -- Indexes for table `product`
432     --
433 ●    ALTER TABLE `product`
434         ADD PRIMARY KEY (`id`),
435         ADD KEY `category_id` (`category_id`);
436
437     --
438     -- Indexes for table `productprice`
439     --
440     --
441     -- Indexes for table `productprice`
442     --
443 ●    ALTER TABLE `productprice`
444         ADD PRIMARY KEY (`id`),
445         ADD KEY `product_id` (`product_id`);
```

Ràng buộc tự tăng:

```
454 ● ALTER TABLE `account`
455     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
456
457     --
458     -- AUTO_INCREMENT for table `blog`
459     --
460 ● ALTER TABLE `blog`
461     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
462
463     --
464     -- AUTO_INCREMENT for table `category`
465     --
466 ● ALTER TABLE `category`
467     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;
468
469 ● ALTER TABLE `image`
470     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=61;
471
472     --
473     -- AUTO_INCREMENT for table `order`
474     --
475     --
476     -- AUTO_INCREMENT for table `orderdetail`
477     --
478 ● ALTER TABLE `order`
479     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;
480
481     --
482     -- AUTO_INCREMENT for table `orderdetail`
483     --
484 ● ALTER TABLE `orderdetail`
485     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;
486
487 ● ALTER TABLE `product`
488     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=61;
489
490     --
491     -- AUTO_INCREMENT for table `productprice`
492     --
493     --
494     -- AUTO_INCREMENT for table `productprice`
495     --
496 ● ALTER TABLE `productprice`
497     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=61;
```

Ràng buộc khóa ngoại:

```

506 •    ALTER TABLE `blog`
507      ADD CONSTRAINT `blog_ibfk_1` FOREIGN KEY (`account_id`) REFERENCES `account` (`id`);
508
509      --
510      -- Constraints for table `image`
511      --
512 •    ALTER TABLE `image`
513      ADD CONSTRAINT `image_ibfk_1` FOREIGN KEY (`product_id`) REFERENCES `product` (`id`),
514      ADD CONSTRAINT `image_ibfk_2` FOREIGN KEY (`blog_id`) REFERENCES `blog` (`id`);
515
516      --
517      -- Constraints for table `order`
518      --
519 •    ALTER TABLE `order`
520      ADD CONSTRAINT `order_ibfk_1` FOREIGN KEY (`account_id`) REFERENCES `account` (`id`);

525 •    ALTER TABLE `orderdetail`
526      ADD CONSTRAINT `orderdetail_ibfk_1` FOREIGN KEY (`product_id`) REFERENCES `product` (`id`),
527      ADD CONSTRAINT `orderdetail_ibfk_2` FOREIGN KEY (`order_id`) REFERENCES `order` (`id`);
528
529      --
530      -- Constraints for table `product`
531      --
532 •    ALTER TABLE `product`
533      ADD CONSTRAINT `product_ibfk_1` FOREIGN KEY (`category_id`) REFERENCES `category` (`id`);
534
535      --
536      -- Constraints for table `productprice`
537      --
538 •    ALTER TABLE `productprice`
539      ADD CONSTRAINT `productprice_ibfk_1` FOREIGN KEY (`product_id`) REFERENCES `product` (`id`);
540 •    COMMIT;

```

5. Kết nối database với Xampp

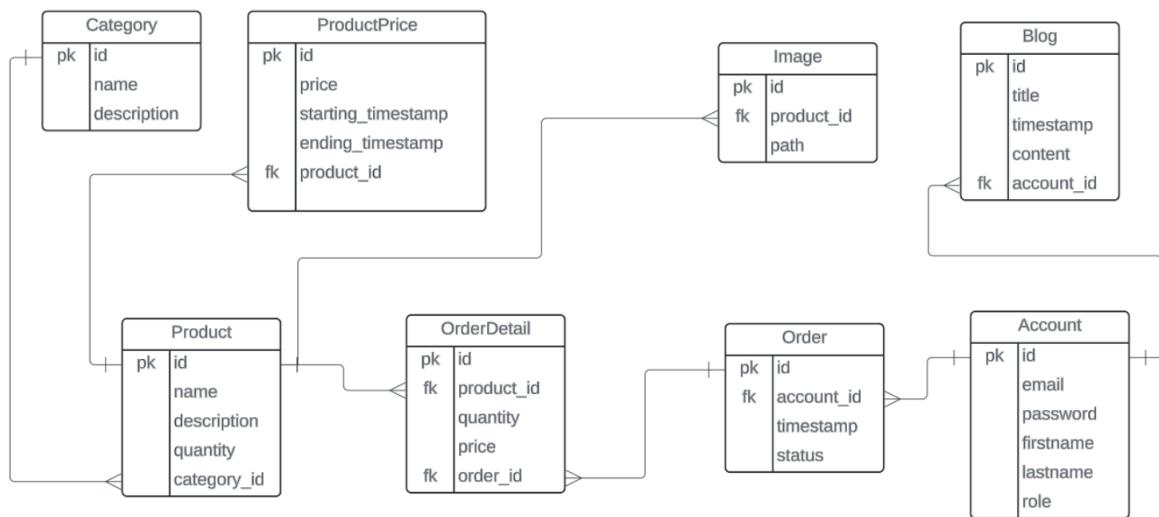
Ta sẽ kết nối database vừa tạo được với Xampp thông qua một file PHP, tạm đặt là db.php như sau:

```

1 <?php
2     $dsn = "mysql:host=localhost;port=3306;dbname=usbibracelet";
3     $dbusername = "root";
4     $dbpassword = "";
5
6     try {
7         $pdo = new PDO($dsn, $dbusername, $dbpassword);
8         $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
9     } catch (PDOException $e) {
10        echo "Connection failed: ". $e->getMessage();
11    }
12 ?>

```

6. ERD



Mô hình thực thể - mối quan hệ của usbibracelet

7. Mô tả mối quan hệ

- Mối quan hệ giữa Category và Product:**

Mối quan hệ: Một category có thể chứa nhiều product .

Loại quan hệ: 1:N (Một-nhiều).

Ý nghĩa: Mỗi sản phẩm (product) thuộc một danh mục (category), nhưng một danh mục có thể chứa nhiều sản phẩm. Mối quan hệ này được thiết lập qua khóa ngoại category_id trong bảng product liên kết với id trong bảng category.

- Mối quan hệ giữa Productprice và Product:**

Mối quan hệ: Một product có thể có nhiều productprice.

Loại quan hệ: 1:N (Một-nhiều).

Ý nghĩa: Mỗi sản phẩm (product) có thể có nhiều mức giá khác nhau theo thời gian, vì vậy một sản phẩm có thể có nhiều bản ghi trong bảng productprice. Mỗi quan hệ này được thiết lập qua khóa ngoại product_id trong bảng productprice liên kết với id trong bảng product.

- **Mối quan hệ giữa Image và Product:**

Mối quan hệ: Một product có thể có nhiều image.

Loại quan hệ: 1:N (Một-nhiều).

Ý nghĩa: Mỗi sản phẩm có thể có nhiều hình ảnh khác nhau, ví dụ như ảnh chính và các ảnh phụ. Mỗi quan hệ này được thiết lập qua khóa ngoại product_id trong bảng image liên kết với id trong bảng product.

- **Mối quan hệ giữa Account và Order:**

Mối quan hệ: Một account có thể có nhiều order.

Loại quan hệ: 1:N (Một-nhiều).

Ý nghĩa: Một người dùng (account) có thể tạo nhiều đơn hàng (order). Mỗi quan hệ này được thiết lập qua khóa ngoại account_id trong bảng order liên kết với id trong bảng account.

- **Mối quan hệ giữa Order và Orderdetail:**

Mối quan hệ: Một order có thể có nhiều orderdetail.

Loại quan hệ: 1:N (Một-nhiều).

Ý nghĩa: Mỗi đơn hàng (order) có thể chứa nhiều chi tiết đơn hàng (orderdetail), ví dụ như các sản phẩm khác nhau trong một đơn hàng. Mỗi quan hệ này được thiết lập qua khóa ngoại order_id trong bảng orderdetail liên kết với id trong bảng order.

- **Mối quan hệ giữa Product và Orderdetail:**

Mối quan hệ: Một product có thể xuất hiện trong nhiều orderdetail.

Loại quan hệ: 1:N (Một-nhiều).

Ý nghĩa: Một sản phẩm có thể xuất hiện trong nhiều chi tiết đơn hàng khác nhau, ví dụ như một sản phẩm có thể được mua trong nhiều đơn hàng khác nhau. Mỗi quan hệ này được thiết lập qua khóa ngoại product_id trong bảng orderdetail liên kết với id trong bảng product.

- **Mối quan hệ giữa Account và Blog:**

Mối quan hệ: Một account có thể tạo nhiều blog.

Loại quan hệ: 1:N (Một-nhiều).

Ý nghĩa: Một tài khoản người dùng (account) có thể viết nhiều bài viết blog (blog). Mối quan hệ này được thiết lập qua khóa ngoại account_id trong bảng blog liên kết với id trong bảng account.

TÀI LIỆU THAM KHẢO

- Duckett, J. (2011). *HTML and CSS: Design and build websites*. Wiley.
- Zeldman, J., & Marcotte, E. (2010). *Designing with web standards* (3rd ed.). New Riders.
- Spool, J. M. (2009). *Web site usability: A designer's guide*. Morgan Kaufmann.

PHỤ LỤC

GitHub: https://github.com/Luw-phuc/DOAN_CK_PTUDTMDT.git