

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO CUỐI KỲ
NHẬP MÔN WEB VÀ ỨNG DỤNG

Tên đề tài:
Website bán gear gaming

HỌ TÊN SV : Thái Tín Khang
LỚP/NHÓM : D21_TH01/Nhóm 1 T7C3
MSSV : DH52102716
GVHD : NGUYỄN TRỌNG NGHĨA

LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy đã tận tình hướng dẫn em hoàn thành Báo cáo đồ án nhập môn web và ứng dụng.

Sự tận tâm, nhiệt tình và kiến thức chuyên môn sâu rộng của thầy đã giúp em rất nhiều trong suốt quá trình thực hiện đồ án. Nhờ những lời khuyên bổ ích và sự động viên của thầy, em đã có thêm động lực để vượt qua những khó khăn và hoàn thành tốt bài tập.

Em cũng xin cảm ơn thầy đã dành thời gian quý báu để giải đáp những thắc mắc của em một cách cẩn thận và chi tiết. Nhờ sự hướng dẫn của thầy, em đã học hỏi được rất nhiều kiến thức mới mẻ và bổ ích về lập trình web và ứng dụng.

Báo cáo đồ án này còn có những thiếu sót do năng lực của em còn hạn chế. Em mong nhận được những góp ý quý báu của thầy để hoàn thiện bản thân trong tương lai.

Cuối cùng, em xin kính chúc thầy sức khỏe dồi dào, hạnh phúc và thành công trong sự nghiệp giảng dạy.

Trân trọng,



MỤC LỤC

I. GIỚI THIỆU	Trang 1
1. Lý do chọn đề tài:	Trang 1
2. Địa chỉ đề tài:	Trang 1
II. NỘI DUNG	Trang 1
1. Giao diện.....	Trang 1
1.1. Các đặc điểm nổi bật của Template	Trang 1
1.2. Bố cục, CSS	Trang 3
1.3. Địa chỉ template tham khảo	Trang 12
2. Các chức năng nổi bật.....	Trang 12
2.1. Chức năng thêm giỏ hàng	Trang 12
2.2. Chức năng hiển thị chi tiết sản phẩm.....	Trang 22
2.3. Chức năng tăng giảm số lượng sản phẩm ở trang chi tiết.....	Trang 28
2.4. Chức năng hiển thị sản phẩm liên quan	Trang 28
2.5. Chức năng hiển thị chi tiết blog.....	Trang 31
2.6. Chức năng upload ảnh hiển thị ảnh preview.....	Trang 34
2.7. Chức năng xoá sản phẩm trong dashboard	Trang 36
2.8. Chức năng xoá danh mục trong dashboard.....	Trang 39
2.9. Chức năng xoá blog trong dashboard	Trang 40
2.10. Hiển thị số lượng sản phẩm theo danh mục sản phẩm	Trang 40
2.11. Hiệu ứng animation placeholder tìm kiếm	Trang 42
2.12. Hiệu ứng zoom highlight	Trang 43
2.13. Hiệu ứng back to top.....	Trang 43
2.14. Hiệu ứng hover sản phẩm	Trang 44
2.15. Carousel banner	Trang 48
2.16. Carousel sản phẩm	Trang 50
2.17. Validate form	Trang 52
2.18. Thư viện đã sử dụng	Trang 57
III. TÀI LIỆU THAM KHẢO	Trang 57

MỤC LỤC HÌNH ẢNH

Hình 1. Trang chủ	Trang 2
Hình 2. Trang chủ	Trang 4
Hình 3. Trang cửa hàng	Trang 5
Hình 4. Trang chi tiết sản phẩm.....	Trang 6
Hình 5. Trang đăng nhập	Trang 7
Hình 6. Trang liên hệ	Trang 7
Hình 7. Trang about	Trang 8
Hình 8. Trang blog.....	Trang 8
Hình 9. Trang chi tiết blog.....	Trang 9
Hình 10. Trang chi tiết giỏ hàng	Trang 9
Hình 11. Trang dashboard	Trang 10
Hình 12. Trang dashboard sản phẩm	Trang 10
Hình 13. Trang dashboard tin tức	Trang 10
Hình 14. Trang dashboard danh mục.....	Trang 11
Hình 15. Trang dashboard thêm sản phẩm	Trang 11
Hình 16. Trang dashboard thêm danh mục.....	Trang 12
Hình 17. Giỏ hàng mini	Trang 12
Hình 18. Hiệu ứng khi thêm sản phẩm vào giỏ hàng	Trang 15
Hình 19. Hiện thị chi tiết giỏ hàng	Trang 16
Hình 20. Xóa sản phẩm khỏi giỏ hàng	Trang 18
Hình 21. Thêm vào giỏ hàng với số lượng sản phẩm	Trang 22
Hình 22. Query params của sản phẩm	Trang 22
Hình 23. Chi tiết sản phẩm query params.....	Trang 26
Hình 24. Tăng giảm số lượng sản phẩm	Trang 28
Hình 25. Sản phẩm liên quan.....	Trang 29
Hình 26. Query param blog	Trang 31
Hình 27. Chi tiết blog query params.....	Trang 33
Hình 28. Trước khi upload ảnh.....	Trang 35
Hình 29. Sau khi upload ảnh.....	Trang 35
Hình 30. Xóa sản phẩm dashboard	Trang 36
Hình 31. Xóa danh mục trong dashboard	Trang 39
Hình 32. Xóa blog trong dashboard.....	Trang 40
Hình 33. Số lượng sản phẩm theo danh mục	Trang 40
Hình 34. Hiệu ứng animation placeholder.....	Trang 42
Hình 35. Hiệu ứng zoom highlight.....	Trang 43
Hình 36. Hiệu ứng back to top.....	Trang 44
Hình 37. Hiệu ứng trước khi hover sản phẩm	Trang 45
Hình 38. Hiệu ứng sau khi hover sản phẩm.....	Trang 46
Hình 39. Carousel banner	Trang 48
Hình 40. Carousel sản phẩm.....	Trang 50
Hình 41. Validate form liên hệ	Trang 52
Hình 42. Gửi thông tin form liên hệ thành công.....	Trang 52
Hình 43. Validate form đăng nhập	Trang 56
Hình 44. Validate form thêm sản phẩm.....	Trang 57
Hình 45. Validate form thêm danh mục	Trang 57

I. GIỚI THIỆU

1. Lý do chọn đề tài:

Thị trường game Việt Nam đang bùng nổ với lượng người chơi ngày càng tăng, kéo theo nhu cầu cao về gear gaming - phụ kiện chơi game thiết yếu. Nắm bắt được xu hướng này, đồ án website bán gear gaming là một lựa chọn đầy tiềm năng, hứa hẹn mang lại nhiều lợi ích và cơ hội phát triển.

Nhu cầu thị trường về gear gaming là vô cùng lớn. Nhu cầu chơi game ngày càng tăng cao, đặc biệt là giới trẻ, dẫn đến nhu cầu về gear gaming như chuột, bàn phím, tai nghe, màn hình, ghế chơi game,... cũng tăng theo.

Bên cạnh nhu cầu thị trường cao, đam mê cũng là một yếu tố quan trọng thúc đẩy em lựa chọn đề tài này. Ngoài ra, xây dựng và quản lý website còn giúp em học hỏi nhiều kiến thức mới về lập trình, marketing, bán hàng,... Với những lợi ích và tiềm năng to lớn như vậy, đồ án website bán gear gaming là một lựa chọn sáng suốt cho mà em chọn.

2. Địa chỉ đề tài:

Link: <https://thaitinkhang.000webhostapp.com/>

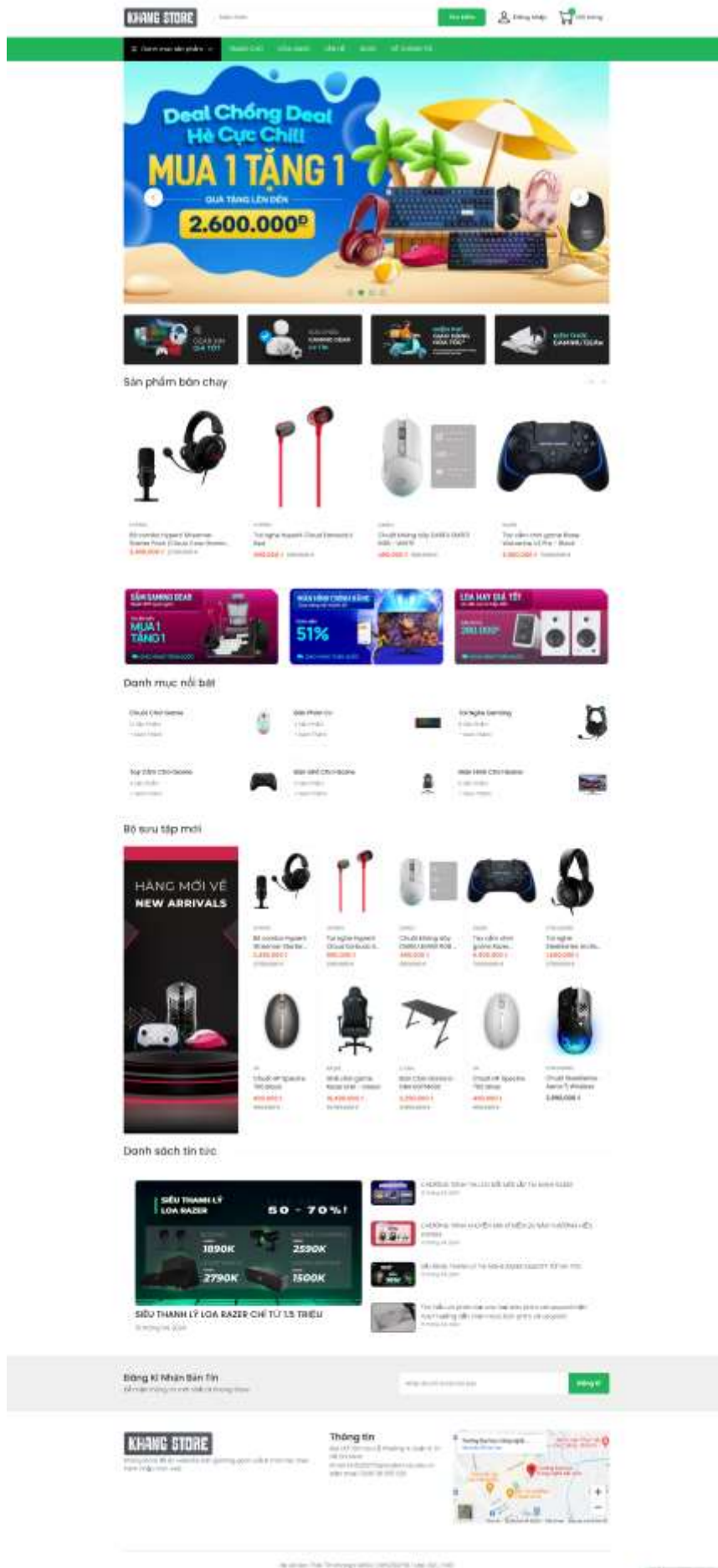
Source: <https://github.com/kinyias/Website-Gaming-Gear-Store>

II. NỘI DUNG

1. Giao diện

1.1. Các đặc điểm nổi bật của Template

- Giao diện hiện đại, dễ sử dụng mang đến trải nghiệm thoải mái, tối ưu hoá UI/UX.
- Giao diện responsive tương thích với nhiều thiết bị (PC, laptop, tablet, mobile,...)
- Có nhiều hiệu ứng như animation placeholder input, back to top, zoom highlight, validate form, hiệu ứng thêm giỏ hàng, ẩn hiện giỏ hàng mini, ẩn hiện danh sách sản phẩm.
- Cho phép hiển thị chi tiết sản phẩm của từng sản phẩm, hiển thị sản phẩm theo danh mục sản phẩm.
- Carousel banner, carousel sản phẩm, responsive carousel cho phép kéo thả carousel.
- Nhúng địa chỉ bản đồ google map.
- Có trang dashboard hiển thị thống kê, thêm sản phẩm, thêm danh mục, tải lên hình ảnh có preview xem trước hình ảnh.



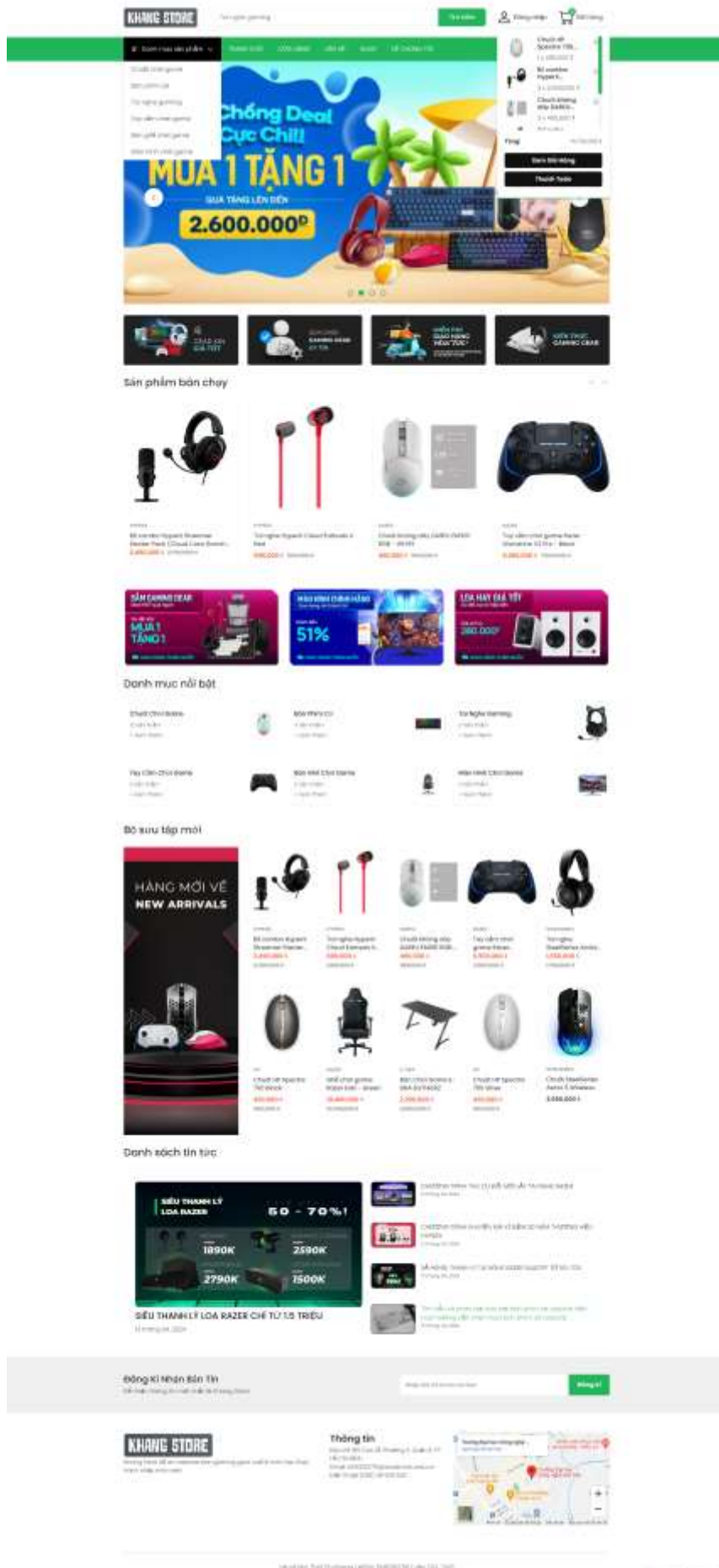
Hình 1. Trang chủ

Đề tài: Website bán gear gaming

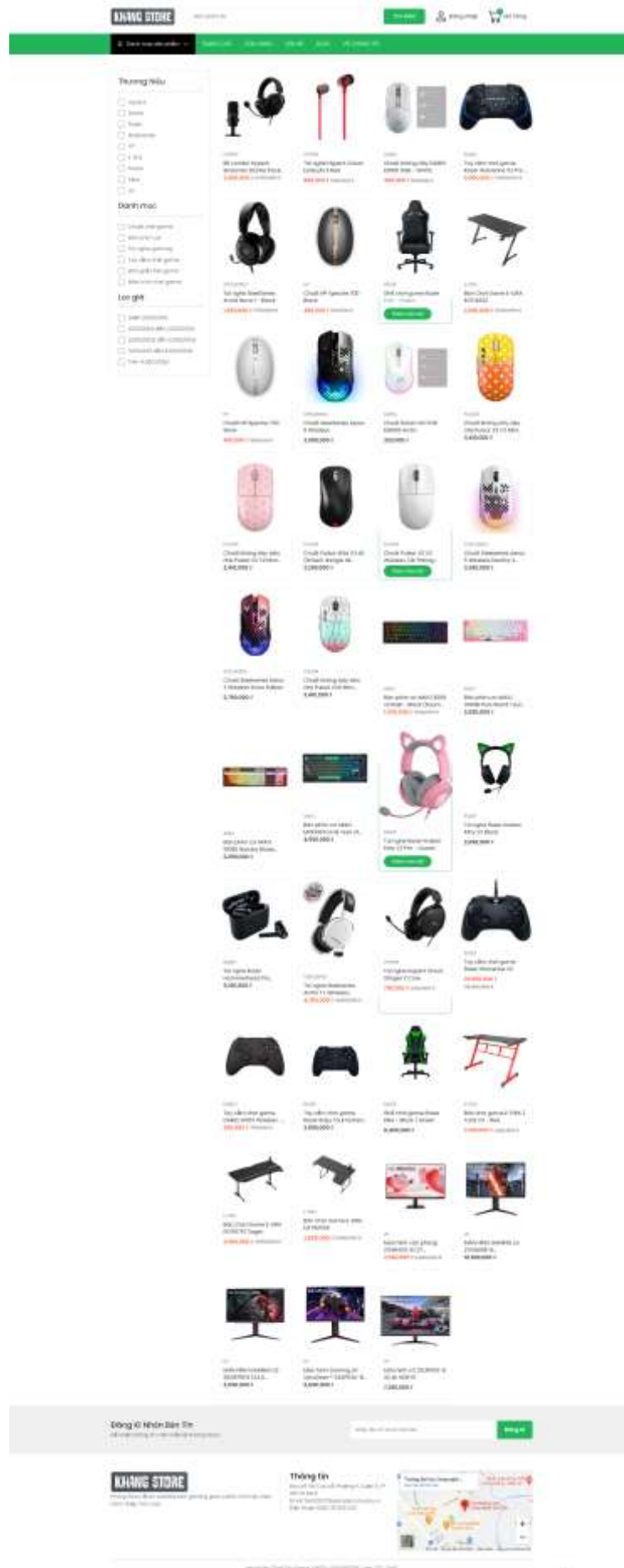
Trang 2

1.2. Bố cục, CSS

- Website gồm 17 trang:
 - + Trang chủ (index.html)
 - + Trang đăng nhập (login.html)
 - + Trang cửa hàng (store.html)
 - + Trang chi tiết sản phẩm (product-detail.html)
 - + Trang về chúng tôi (about.html)
 - + Trang blog (blog.html)
 - + Trang chi tiết blog (blog-detail.html)
 - + Trang chi tiết giỏ hàng (cart-detail.html)
 - + Trang liên hệ (contact.html)
 - + Trang dashboard (dashboard.html)
 - + Trang thêm sản phẩm (dashboard-add-product.html)
 - + Trang chỉnh sửa sản phẩm (dashboard-edit-product.html)
 - + Trang thêm danh mục (dashboard-add-category.html)
 - + Trang chỉnh sửa danh mục (dashboard-edit-category.html)
 - + Trang dashboard danh sách sản phẩm (dashboard-products.html)
 - + Trang dashboard danh sách danh mục (dashboard-categories.html)
 - + Trang dashboard danh sách blog. (dashboard-blogs.html)



Hình 2. Trang chủ



Hình 3. Trang cửa hàng

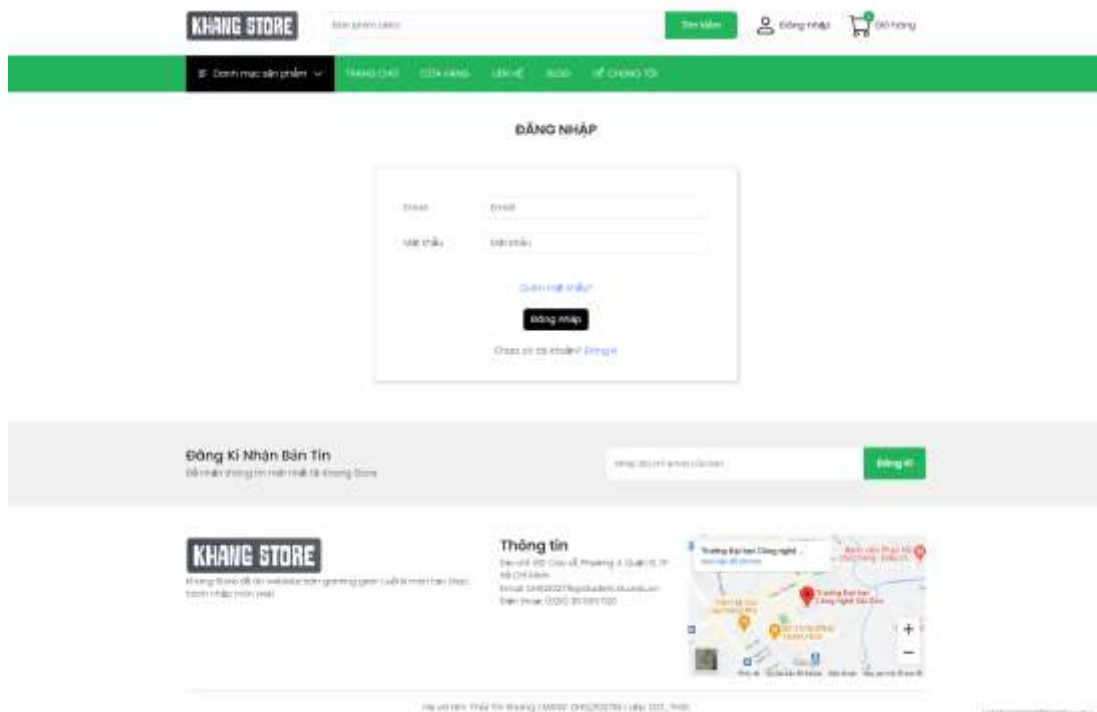
Đề tài: Website bán gear gaming

Trang 5



Hình 4. Trang chi tiết sản phẩm

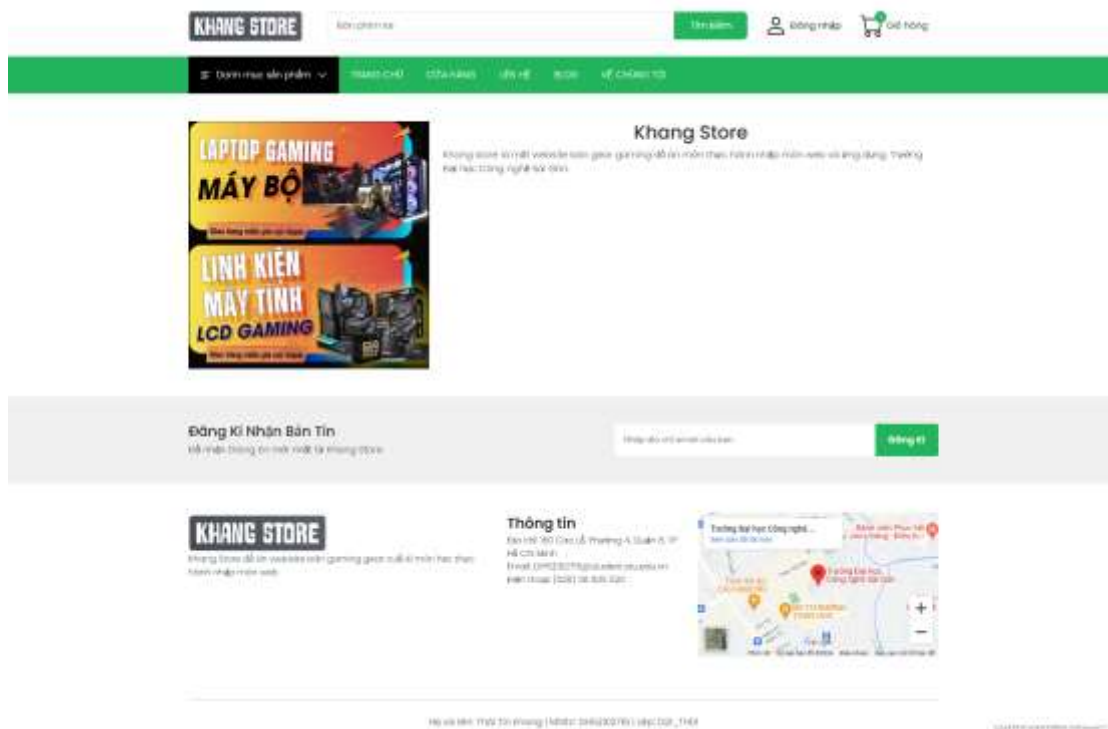
Đề tài: Website bán gear gaming



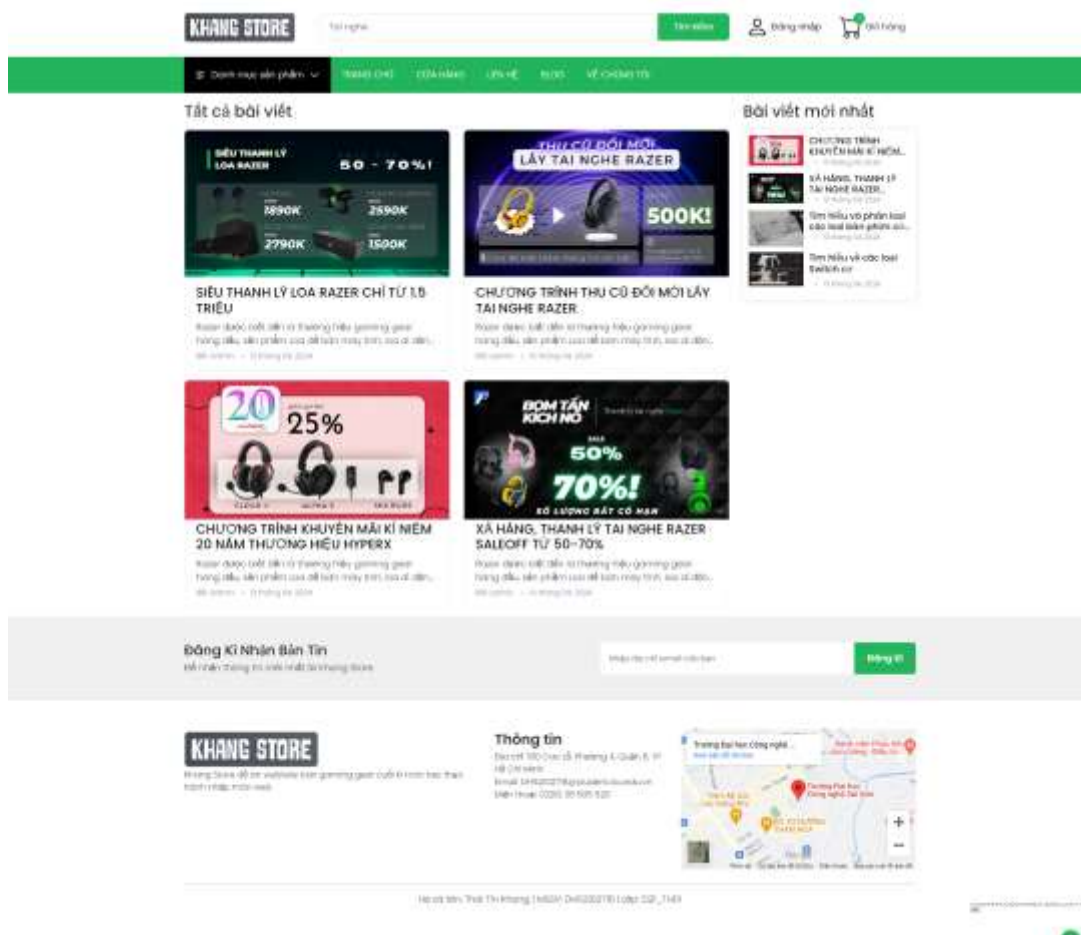
Hình 5. Trang đăng nhập



Hình 6. Trang liên hệ



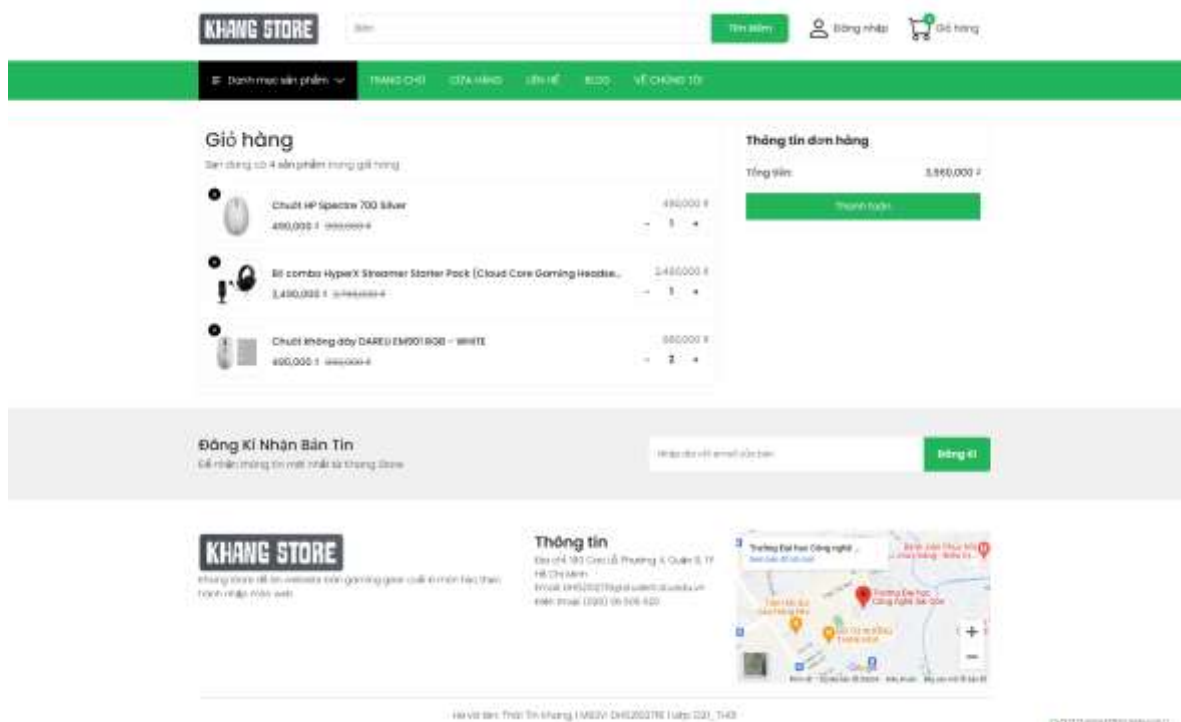
Hình 7. Trang about



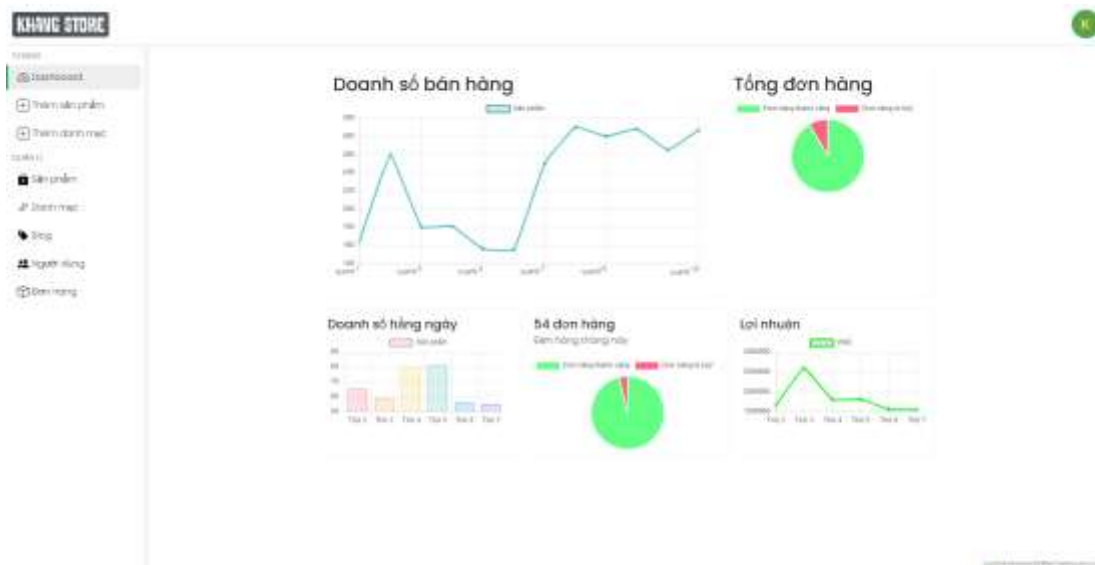
Hình 8. Trang blog



Hình 9. Trang chi tiết blog



Hình 10. Trang chi tiết giỏ hàng



Hình 11. Trang dashboard

The product dashboard for KHANG STORE displays a list of products with their details and a sidebar menu:

Hình ảnh	Tên sản phẩm	Giá	Thêm sản phẩm
	Bộ combo HyperX Elitestar Starter Pack (Cloud Core Gaming Headset QTS + SoloCast USB Microphone)	3,400,000 ₫ (3400000-4)	
	Tai nghe HyperX Cloud Core II Red	690,000 ₫ (690000-4)	
	Chuột không dây OMEN OM901 RGB - WHITE	490,000 ₫ (490000-4)	
	Tay cầm chơi game Razer Wolverine V2 Pro - Black	6,900,000 ₫ (6900000-4)	
	Tai nghe SteelSeries Arctis Nova 1 - Black	1,000,000 ₫ (1000000-4)	
	Chuột HP Spectre 700 Black	490,000 ₫ (490000-4)	

The sidebar menu includes the following items:

- Dashboard
- Thêm sản phẩm
- Thêm danh mục
- Quản lý
- Sản phẩm
- Đơn hàng
- Blog
- Người dùng
- Đơn hàng

Hình 12. Trang dashboard sản phẩm

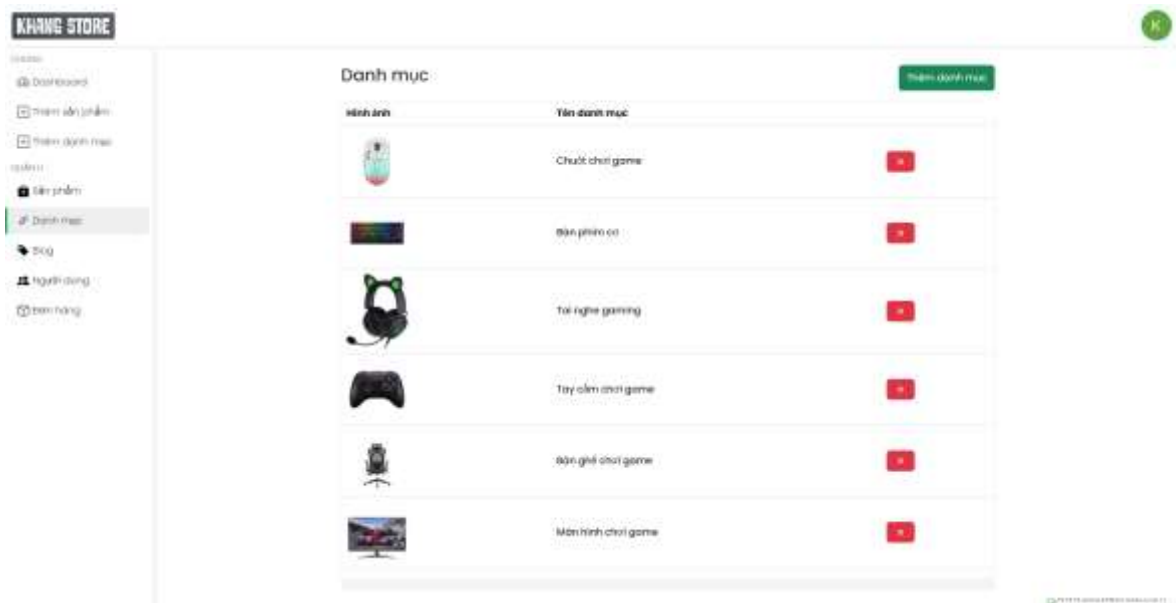
The news dashboard for KHANG STORE displays a list of news items with their details and a sidebar menu:

Hình ảnh	Tiêu đề	Thêm tin tức
	SIÊU THÀNH LẬP HUA RAZER CHỈ TỪ 15 TRIỆU	
	CHƯƠNG TRÌNH THƯ CỐ ĐÓNG LẮC TẠI NGÔI RAZER	
	CHƯƠNG TRÌNH KHUYẾN MÃI KẾT HỢP 20 NĂM THƯƠNG HIỆU HYPERX	
	XÃ HỘI, THÀNH LẬP TÀI NGUYÊN RAZER SAU CHỖ TỰ SÁO 2025	
	Tin tức về phân loại các loại bán phẩm có computer bán máy? Hướng dẫn chọn mua bán phẩm có computer	
	Tin tức về các loại bán phẩm có	

The sidebar menu includes the following items:

- Dashboard
- Thêm sản phẩm
- Thêm danh mục
- Quản lý
- Sản phẩm
- Đơn hàng
- Blog
- Người dùng
- Đơn hàng

Hình 13. Trang dashboard tin tức



Hình 14. Trang dashboard danh mục



Hình 15. Trang dashboard thêm sản phẩm



Hình 16. Trang dashboard thêm danh mục

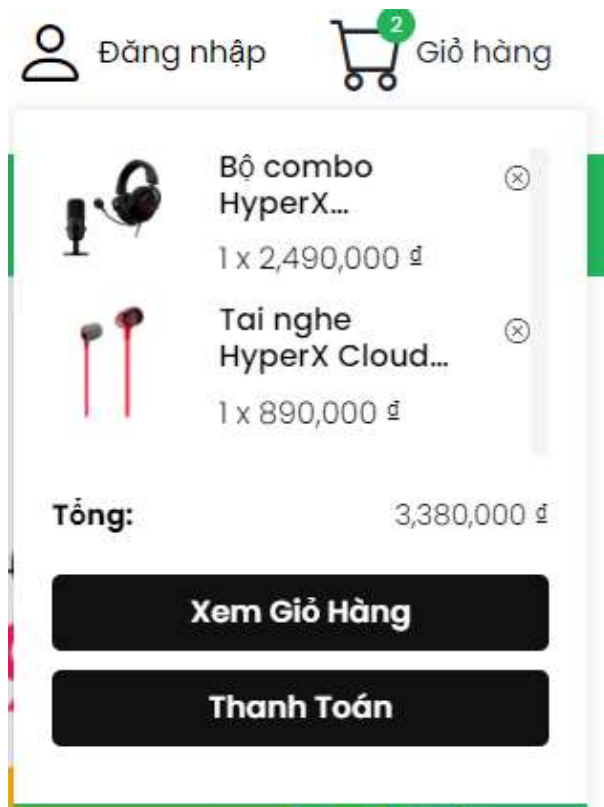
1.3. Địa chỉ template tham khảo

Link: https://preview.themeforest.net/item/sinrato-mega-shop-electronics-ecommerce-bootstrap-4-template/full_screen_preview/22738390

2. Các chức năng nổi bật

2.1. Chức năng thêm giỏ hàng

- Cho phép bấm nút “thêm vào giỏ” thêm sản phẩm vào giỏ hàng, hiển thị số lượng sản phẩm và tự động tính tổng giá tiền của tất cả sản phẩm:



Hình 17. Giỏ hàng mini

+JS:

```
/**
 * Hàm này để thêm sản phẩm vào giỏ hàng
 * @param productId
 */
function addToCart(productId) {
  let cart = JSON.parse(localStorage.getItem('cart')); //Lấy sản phẩm từ localStorage
  positionInCart = cart.findIndex((item) => item.productId == productId); //Tìm vị trí của sản phẩm trong giỏ hàng
  const input_quantity = document.getElementById('quantity');
  const quantity = input_quantity ? Number(input_quantity.value) : 1;
  //Nếu có input số lượng sp thì lấy số lượng sản phẩm của input còn không thì số lượng sp = 1
  if (cart.length <= 0) {
    //Nếu giỏ hàng chưa có sản phẩm nào thì thêm sp vào cart
    cart.push({
      productId: productId,
      quantity: quantity,
    });
  } else if (positionInCart < 0) {
    //Nếu không tìm thấy sp nào trùng với sp được thêm thì thêm sp vào cart
    cart.push({
      productId: productId,
      quantity: quantity,
    });
  } else {
    cart[positionInCart].quantity += quantity; // Tìm thấy sp trong giỏ hàng thì số lượng sp + quantity
  }
  localStorage.setItem('cart', JSON.stringify(cart)); //Lưu cart vào localStorage
  loadCartToHTML(); //Render lại giao diện
}

/**
 * Hàm này dùng để render ra giao diện giỏ hàng
 */
function loadCartToHTML() {
  if (!localStorage.getItem('cart')) {
    //Khởi tạo cart nếu chưa có
    localStorage.setItem('cart', '[]');
  }
  let cart = JSON.parse(localStorage.getItem('cart'));
  const cart_count = document.querySelector('.cart-count');
  //Tổng số sp trong giỏ hàng
  cart_count.textContent = cart.reduce((prev, current) => {
    return prev + current.quantity;
  }, 0);
  const cart_list = document.querySelector('.cart-list');
```

```

const total_price = document.querySelector('.total-price');
let sum = 0;
cart_list.innerHTML = '';
if (cart.length <= 0) {
    //Chưa thêm vào giỏ hàng thì xuất hiện thị
    cart_list.textContent = 'Bạn chưa thêm sản phẩm';
}
//Render HTML giỏ hàng
cart.forEach((item) => {
    let product = getProductById(item.productId);
    let newHTML = `<li class="cart-item">
        <div class="cart-image">
            <a href="product-detail.html?id=${product.id}">
                
            </a>
        </div>
        <div class="cart-info">
            <h4><a href="product-detail.html?id=${product.id}">${product.name}
        </a></h4>
            <span
                >${item.quantity} x
                <span>${formatVND(product.price)}</span>
            </span>
        </div>
        <div class="del-icon" onclick="deleteCart(${item.productId})">
            <i class="bi bi-x-circle"></i>
        </div>
    </li>`;
    cart_list.innerHTML += newHTML;
    sum += item.quantity * product.price;
});
total_price.textContent = formatVND(sum);
}

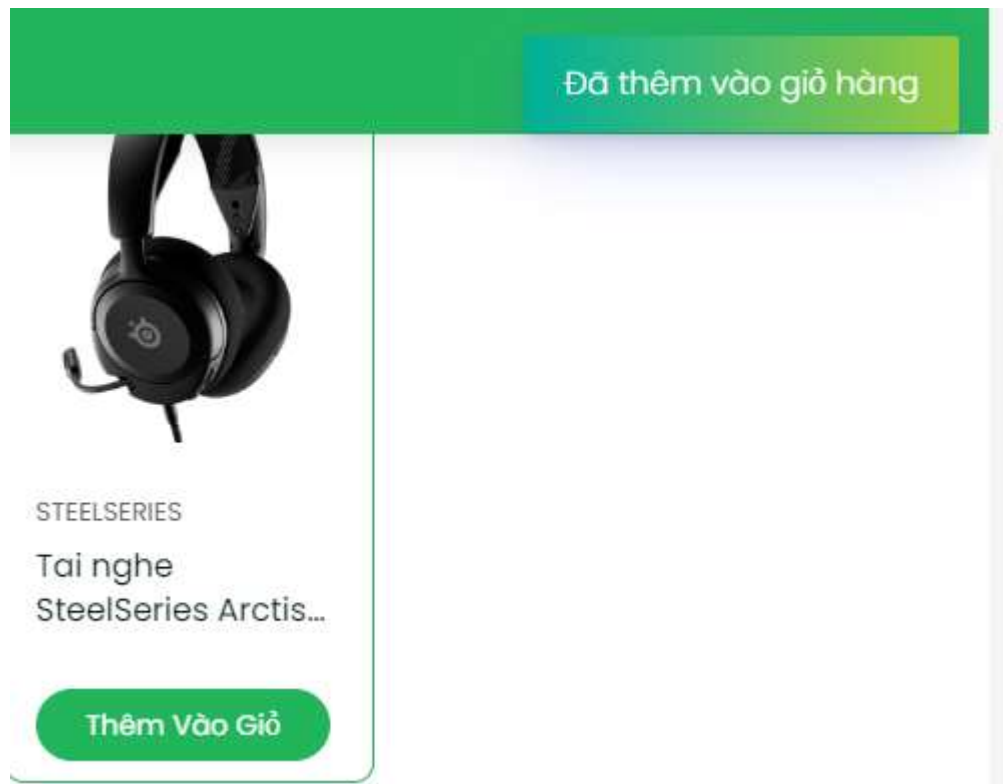
/**
 * Hàm này dùng để format giá từ number -> tiền tệ VN
 * @param number
 * @returns formatted_number
 */
function formatVND(number) {
    let formatted_number = new Intl.NumberFormat('vi-VN', {
        style: 'currency',
        currency: 'VND',
    })
    .format(number)
    .replaceAll('.', ',');
    return formatted_number;
}

```

Mô tả:

Hàm addToCart(productId) được thực hiện mỗi khi click nút thêm giỏ hàng sau khi thêm giỏ hàng và lưu vào localStorage thì thực hiện render lại giao diện chi tiết giỏ hàng.

- Hiệu ứng khi nhấn nút “thêm vào giỏ”:



Hình 18. Hiệu ứng khi thêm sản phẩm vào giỏ hàng

+JS:

```
/**
 * Hàm này dùng để show toast message mỗi khi thêm sp vào giỏ
 */
function toastMessage() {
  const btn_carts = document.querySelectorAll('.btn-cart');
  btn_carts.forEach((btn) => {
    btn.addEventListener('click', () => {
      Toastify({ //Gọi thư viện
        text: 'Đã thêm vào giỏ hàng',
        style: {
          background: 'linear-gradient(to right, #00b09b, #96c93d)',
        },
        duration: 3000,
      }).showToast();
    });
  });
}
```

- Hiển thị chi tiết giỏ hàng và cho phép thay đổi số lượng sản phẩm tự động tính lại tổng tiền sau khi thay đổi số lượng số lượng sản phẩm:



Hình 19. Hiển thị chi tiết giỏ hàng

```
/**
 * Hàm này để tăng số lượng sp trong giỏ hàng
 * @param id
 */
function minusQuantity(id) {
    let cart = JSON.parse(localStorage.getItem('cart')); // Lấy tất cả
    sp trong giỏ hàng từ localStorage
    if (cart.findIndex((item) => item.productId == id).quantity >
    1)
        //Tìm sp trong giỏ hàng và -1 nếu >1 vì không có sp nào trong giỏ
        hàng lại <1
        cart[cart.findIndex((item) => item.productId == id)].quantity -=
    1;
    localStorage.setItem('cart', JSON.stringify(cart));
    loadCartDetail(); //Render lại giao diện chi tiết giỏ hàng
}

/**
 * Hàm này để giảm số lượng sp trong giỏ hàng
 * @param id
 */
function plusQuantity(id) {
    let cart = JSON.parse(localStorage.getItem('cart')); // Lấy tất cả sp
    trong giỏ hàng từ localStorage
    cart[cart.findIndex((item) => item.productId == id)].quantity += 1;
    //Tìm sp trong giỏ hàng và +1
    localStorage.setItem('cart', JSON.stringify(cart));
    loadCartDetail(); //Render lại giao diện chi tiết giỏ hàng
}

/**
 * Hàm này để render giao diện chi tiết giỏ hàng
 */
function loadCartDetail() {
```

```

if (!localStorage.getItem('cart')) {
    //Khởi tạo cart nếu chưa có
    localStorage.setItem('cart', '[]');
}
let cart = JSON.parse(localStorage.getItem('cart'));
const table_cart = document.querySelector('.table-cart');
const title_number_cart = document.querySelector('.title-number-cart');
title_number_cart.innerHTML = `Bạn đang có <strong> ${cart.reduce(
    (prev, current) => {
        return prev + current.quantity;
    },
    0
)} sản phẩm </strong> trong giỏ hàng`;
const summary_total_number = document.querySelector('.summary-total-number');
let sum = 0;
table_cart.innerHTML = '';
//Render HTML
cart.forEach((item) => {
    let product = getProductById(item.productId);
    let newHTML = `<div class="media-line-item line-item">
        <div class="media-left">
            <div class="item-img">
                <a href="/product-detail.html?id=${product.id}"> </a>

            </div>
            <div class="item-remove"
                onclick="confirmDelete(${item.productId})">
                <i class="bi bi-x"></i>
            </div>
        </div>
        <div class="media-right">
            <div class="item-info">
                <a href="/product-
detail.html?id=${product.id}">${product.name}</a>
            </div>
            <div class="item-price">
                <span>${formatVND(product.price)}</span>
                <del>${product.price_old ? formatVND(product.price_old) :
''}</del>
            </div>
        </div>
        <div class="media-total">
            <div class="item-total-price">
                <span>${formatVND(item.quantity * product.price)}</span>
            </div>
        </div>
    `;
    table_cart.innerHTML += newHTML;
    sum += item.quantity * product.price;
});
summary_total_number.innerHTML = `Tổng cộng: ${formatVND(sum)} VND`;

```

```

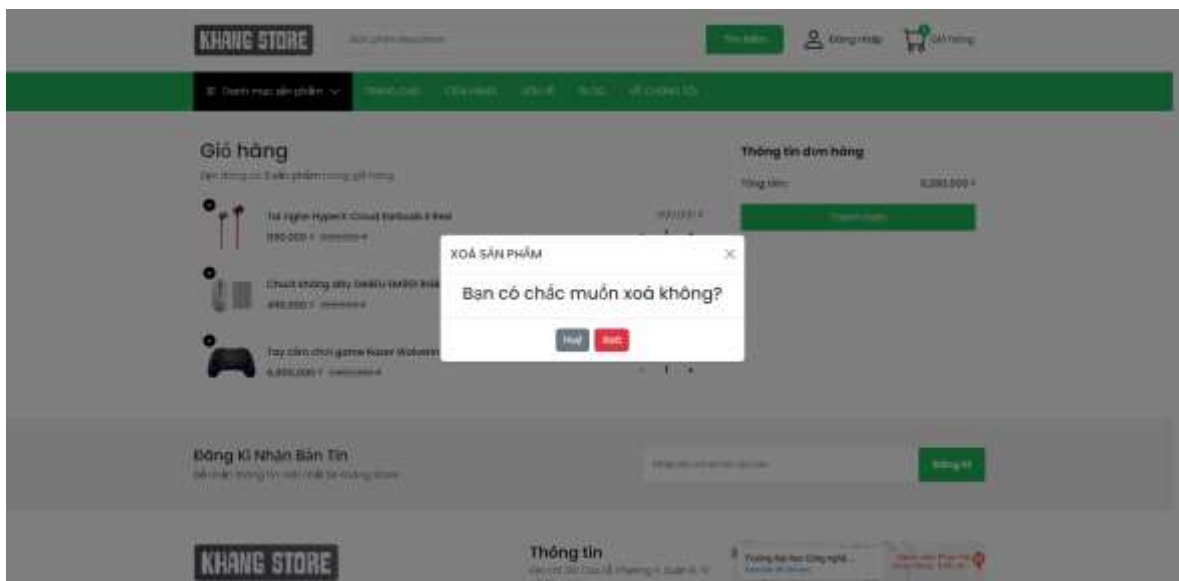
<div class="item-qty">
  <div class="qty">
    <button type="button" onclick="minusQuantity({{
      item.productId
    }}" class="btn-qty">-</button>
    <input
      type="text"
      class="quantity-input"
      value="{{ item.quantity }}"
      min="1"
    />
    <button type="button" onclick="plusQuantity({{
      item.productId
    }}" class="btn-qty">+</button>
  </div>
</div>
</div>`;
table_cart.innerHTML += newHTML;
sum += item.quantity * product.price;
});
summary_total_number.textContent = formatVND(sum);
}

```

Mô tả:

Mỗi khi thực hiện tăng hoặc giảm sản phẩm sẽ thực hiện hàm để tính toán và render lại giao diện cho phù hợp

- Có thể xóa sản phẩm khỏi giỏ hàng bằng cách nhấn nút “x”. Và hiển thị modal xác nhận trước khi xóa.



Hình 20. Xóa sản phẩm khỏi giỏ hàng

+HTML:

```

<!-- Button trigger modal -->
<button
  type="button"
  id="btnModal"

```

```

        style="display: none"
        class="btn btn-primary"
        data-bs-toggle="modal"
        data-bs-target="#modal"
    >
        Launch demo modal
</button>

<!-- Modal -->
<div
    class="modal fade"
    id="modal"
    tabindex="-1"
    aria-labelledby="modalLabel"
    aria-hidden="true"
>
    <div class="modal-dialog modal-dialog-centered">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5" id="modalLabel">XOÁ SẢN
PHẨM</h1>
                <button
                    type="button"
                    class="btn-close"
                    data-bs-dismiss="modal"
                    aria-label="Close"
                ></button>
            </div>
            <div
                class="modal-body w-100 d-flex justify-content-center
align-items-center flex-column"
            >
                <h3>Bạn có chắc muốn xoá không?</h3>
            </div>
            <div class="modal-footer w-100 d-flex justify-content-
center">
                <button
                    type="button"
                    class="btn btn-secondary"
                    data-bs-dismiss="modal"
                >
                    Huỷ
                </button>
                <button
                    type="button"
                    class="btn btn-danger"
                    id="btn-delete"
                    data-bs-dismiss="modal"
                >
                    Xoá

```

```

        </button>
      </div>
    </div>
  </div>
</div>

```

+JS:

```

/**
 * Hàm này để xác nhận trước khi xóa khỏi giỏ hàng
 * @param id
 */
function confirmDelete(id) {
  const btn_modal = document.getElementById('btnModal');
  btn_modal.click(); //Thực hiện click button modal để hiện modal của
bootstrap
  const btn_delete = document.getElementById('btn-delete'); //Đây là
nút xóa trong modal
  btn_delete.onclick = function () {
    deleteCart(id); //Thực hiện xóa sản phẩm khỏi giỏ hàng
  };
}
/**
 * Xóa sản phẩm khỏi giỏ hàng
 * @param id
 */
function deleteCart(id) {
  let cart = JSON.parse(localStorage.getItem('cart')); //Lấy giỏ hàng
từ localStorage
  cart = cart.filter((item) => item.productId !== id); //Lọc bỏ sản
phẩm cần xóa khỏi giỏ hàng
  localStorage.setItem('cart', JSON.stringify(cart)); //Lưu lại giỏ
hàng vào localStorage
  loadCartDetail(); //Render lại giao diện chi tiết giỏ hàng
}
/**
 * Hàm này để render giao diện chi tiết giỏ hàng
 */
function loadCartDetail() {
  if (!localStorage.getItem('cart')) {
    //Khởi tạo cart nếu chưa có
    localStorage.setItem('cart', '[]');
  }
  let cart = JSON.parse(localStorage.getItem('cart'));
  const table_cart = document.querySelector('.table-cart');
  const title_number_cart = document.querySelector('.title-number-
cart');
  title_number_cart.innerHTML = `Bạn đang có <strong> ${cart.reduce(
    (prev, current) => {
      return prev + current.quantity;
    },

```



```

    0
  }) sản phẩm </strong> trong giỏ hàng`;
  const summary_total_number = document.querySelector('.summary-
total-number');
  let sum = 0;
  table_cart.innerHTML = '';
  //Render HTML
  cart.forEach((item) => {
    let product = getProductById(item.productId);
    let newHTML = `<div class="media-line-item line-item">
      <div class="media-left">
        <div class="item-img">
          <a href="/product-detail.html?id=${product.id}"> </a>

          </div>
          <div class="item-remove"
            onclick="confirmDelete(${item.productId})">
            <i class="bi bi-x"></i>
          </div>
        </div>
        <div class="media-right">
          <div class="item-info">
            <a href="/product-
detail.html?id=${product.id}">${product.name}</a>
          </div>
          <div class="item-price">
            <span>${formatVND(product.price)}</span>
            <del>${product.price_old ? formatVND(product.price_old) :
''}</del>
          </div>
        </div>
        <div class="media-total">
          <div class="item-total-price">
            <span>${formatVND(item.quantity * product.price)}</span>
          </div>
          <div class="item-qty">
            <div class="qty">
              <button type="button" onclick="minusQuantity(${
                item.productId
              })" class="btn-qty">-</button>
              <input
                type="text"
                class="quantity-input"
                value="${item.quantity}"
                min="1"
              />
              <button type="button" onclick="plusQuantity(${

```

```

        item.productId
      })" class="btn-qty">+</button>
    </div>
  </div>
</div>`;
table_cart.innerHTML += newHTML;
sum += item.quantity * product.price;
});
summary_total_number.textContent = formatVND(sum);
}

```

Mô tả:

Hàm `confirmDelete(id)` được gọi mỗi khi click vào dấu “x” và xuất hiện modal xác nhận trước khi xoá. Nếu nhấn nút “xoá” sẽ thực hiện hàm `deleteCart(id)` và tắt modal, nếu nhấn nút “huỷ” thì không làm gì cả và tắt modal

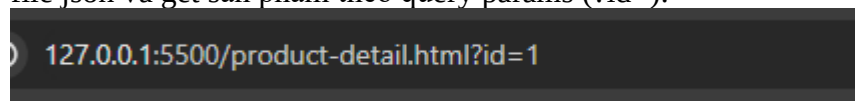
- Cho phép thêm vào giỏ hàng với số lượng sản phẩm đã chọn



Hình 21. Thêm vào giỏ hàng với số lượng sản phẩm

2.2. Chức năng hiển thị chi tiết sản phẩm

- Cho phép hiển thị chi tiết sản phẩm theo từng sản phẩm bằng cách sử dụng data file json và get sản phẩm theo query params (?id=).



Hình 22. Query params của sản phẩm

+JS:

```

function initApp() {
  //Fetch dữ liệu từ file json
  const request1 = fetch('./data/Products.json').then((response) =>
    response.json()
  );
  const request2 = fetch('./data/Categories.json').then((response) =>
    response.json()
  );
}

```

```

);
const request3 = fetch('./data/Blogs.json').then((response) =>
  response.json()
);
Promise.all([request1, request2, request3])
  .then(([data1, data2, data3]) => {
    listProducts = data1;
    listCategories = data2;
    listBlogs = data3;
    generateSearchKey(); //Tạo animation placeholder
    generateCategories(listCategories); // render list danh mục sản phẩm
    generatefeaturedCategoriesList(listCategories); // render danh mục sản phẩm nổi bật
    generateProductList(listProducts); //render list sản phẩm
    generateCollections(listProducts); // render bộ sưu tập sản phẩm
    generateBlogs(listBlogs); //render blogs
    toastMessage();
    loadCartToHTML();
  })
  .catch((error) => {
    console.error(error);
  });
}
/**
 * Hàm này dùng để render list sản phẩm
 * @param listProducts
 */
function generateProductList(listProducts) {
  const productList = document.querySelector(
    '.product-list .owl-stage-outer .owl-stage'
  );
  if (productList) {
    const listShow = listProducts.slice(0, 5);
    listShow.forEach((item) => {
      productList.innerHTML += `<div class="owl-item">
        <div class="product-item">
          <div class="product-thumb">
            <a href="product-detail.html?id=${item.id}">
              
            </a>
          </div>
          <div class="product-caption">
            <div class="manufacture-product">
              <a href="product-detail.html?id=${item.id}">${item.brand}</a>
            </div>
            <div class="product-name">

```

```

<a href="product-detail.html?id=${item.id}">
<h4>
${item.name}
</h4>
</a>
</div>
<div class="price-box">
    <span class="regular-price ${
        item.price_old ? 'sale' : ''
    }">${formatVND(item.price)}</span>
    <span class="old-price">${
        item.price_old ? formatVND(item.price_old) : ''
    }</span>
</div>
<button class="btn-cart" onclick="addToCart(${
    item.id
})" type="button">
    Thêm vào giỏ
</button>
</div>
</div>`;
});
//Sử dụng thư viện owl-carousel
$(document.querySelector('.product-list')).owlCarousel({
    loop: true,
    margin: 30,
    singleItem: true,
    items: 4,
    dots: false,
    nav: true,
    navText: [
        '<i class="bi bi-arrow-left"></i>',
        '<i class="bi bi-arrow-right"></i>',
    ],
    navContainer: '.box-title .custom-nav-best-seller',
    responsive: {
        0: {
            items: 2,
        },
        600: {
            items: 2,
        },
        1000: {
            items: 4,
        },
    },
});
}
}

```

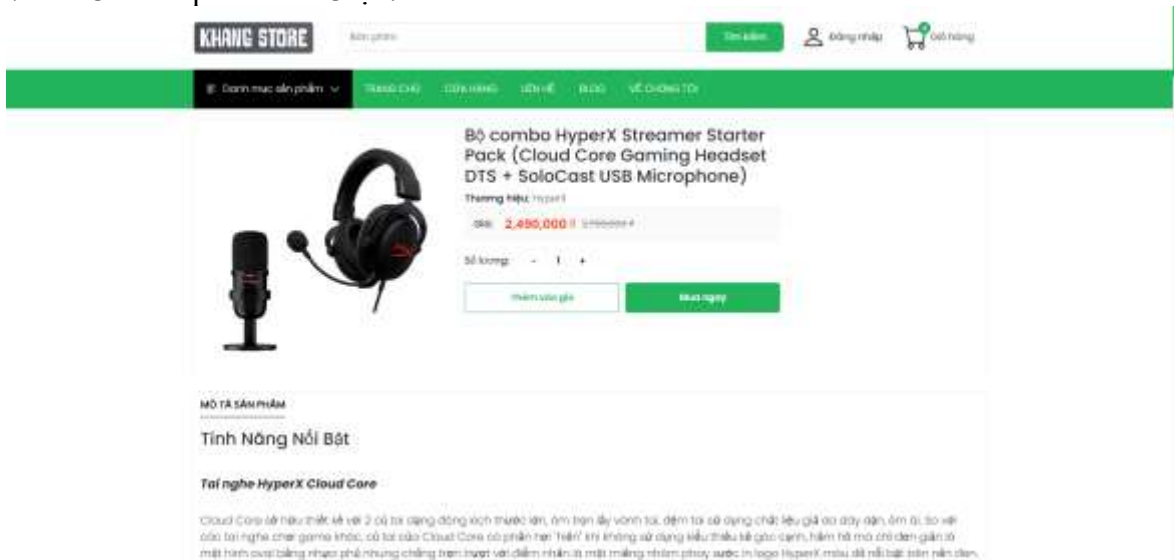
```

/**
 * Hàm này dùng để render list bộ sưu tập sản phẩm
 * @param listProducts
 */
function generateCollections(listProducts) {
  const collections = document.querySelector('.collections-list');
  if (collections) {
    const listShow = listProducts.slice(0, 10);
    listShow.forEach((item) => {
      collections.innerHTML += `<li class="collections-item">
        <div class="product-item">
          <div class="product-thumb">
            <a href="product-detail.html?id=${item.id}">
              
            </a>
          </div>
          <div class="product-caption">
            <div class="manufacture-product">
              <a hrefproduct-
detail.html?id=${item.id}">${item.brand}</a>
            </div>
            <div class="product-name">
              <a href="product-detail.html?id=${item.id}">
                <h4>
                  ${item.name}
                </h4>
              </a>
            </div>
            <div class="price-box">
              <span class="regular-price ${
                item.price_old ? 'sale' : ''
              }">${formatVND(item.price)}</span>
              <span class="old-price">${
                item.price_old ? formatVND(item.price_old) : ''
              }</span>
            </div>
            <button class="btn-cart" onclick="addToCart(${
              item.id
            })" type="button">
              Thêm vào giỏ
            </button>
          </div>
        </div>
      </li>`;
    });
  }
}

```

Mô tả:

Dữ liệu sản phẩm được fetch từ file Products.json và render ra giao diện bằng các hàm generateProducts và generateCollections trong đó thẻ a có thuộc href là “product-detail.html?id=\${item.id}” sẽ đưa người dùng đến trang chi tiết sản phẩm có id tương ứng với id của sản phẩm vừa chọn.



Hình 23. Chi tiết sản phẩm query params

+JS:

```
function initProductDetail() {
  //Fetch dữ liệu từ file json
  fetch('./data/Products.json')
    .then((response) => response.json())
    .then((response) => {
      listProducts = response;
      generateProductByIdQueryParam();
    });
}
/**
 * Hàm này dùng để render sản phẩm theo id params
 */
function generateProductByIdQueryParam() {
  //get id params
  const params = new Proxy(new
  URLSearchParams(window.location.search), {
    get: (searchParams, prop) => searchParams.get(prop),
  });
  let idProduct = params.id;
  let product = listProducts.filter((p) => p.id == idProduct)[0];
  //Lọc ra sản phẩm cần tìm
  const product_detail = document.querySelector('.product-detail');
  const description_content = document.querySelector('.description-
  content');
  //render HTML
  product_detail.innerHTML = `<div class="row">
  <div class="col-lg-4 col-12">
    <div class="product-detail-image">
      
```

```

    </div>
</div>
<div class="col-lg-5 col-12">
    <div class="product-detail-content">
        <div class="product-detail-name">
            <h3>${product.name}</h3>
        </div>
        <div class="product-detail-brand">
            <strong style="font-weight: 600">Thương hiệu:</strong>
            ${product.brand}
        </div>
        <div class="product-detail-price">
            <span class="price-title">Giá: </span>
            <span class="price ${product.price_old ? 'sale' :
            ''}>${formatVND(
            product.price
            )}</span>
            <del class="price-old">${
            product.price_old ? formatVND(product.price_old) : ''
            }</del>
        </div>
        <div class="product-detail-actions">
            <div class="product-detail-quantity">
                <div class="quantity-title">Số lượng:</div>
                <button class="btn-qty" onclick="minusProduct()">-</button>
                <input
                    type="text"
                    id="quantity"
                    name="quantity"
                    value="1"
                    min="1"
                    class="quantity-input"
                />
                <button class="btn-qty" onclick="plusProduct()">+</button>
            </div>
            <div class="product-detail-addToCart mt-3">
                <button class="btn btn-cart" onclick="addToCart(${
                    product.id
                })">Thêm vào giỏ</button>
                <button class="btn btn-buynow"><a href="cart-
                detail.html">Mua ngay</a></button>
            </div>
        </div>
    </div>
</div>`;
    description_content.innerHTML = product.description;
    generateProductRelateList(listProducts, product.category,
    product.id);
    toastMessage();

```

```

mediumZoom('.zoom', {
  margin: 50,
  background: '#000',
});
}

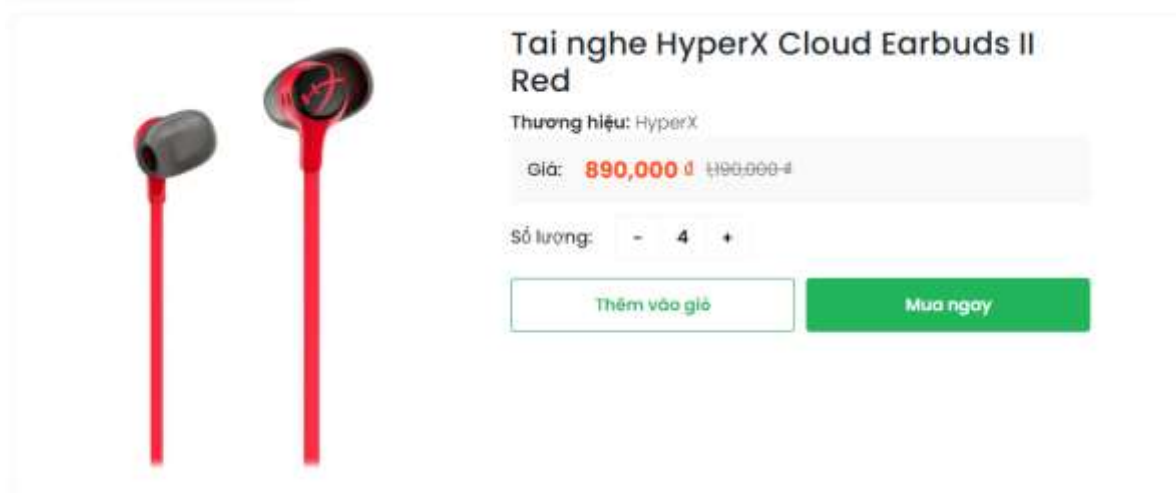
```

Mô tả:

Đầu tiên fetch dữ liệu từ file Products.json sau đó thực hiện hàm generateProductByIdParam lấy id query params từ url sau đó thực hiện lọc dữ liệu file Products.json trùng với id params và thực hiện render giao diện chi tiết sản phẩm

2.3. Chức năng tăng giảm số lượng sản phẩm ở trang chi tiết

- Có 2 nút tăng giảm để tăng giảm số lượng sản phẩm.



Hình 24. Tăng giảm số lượng sản phẩm

+JS:

```

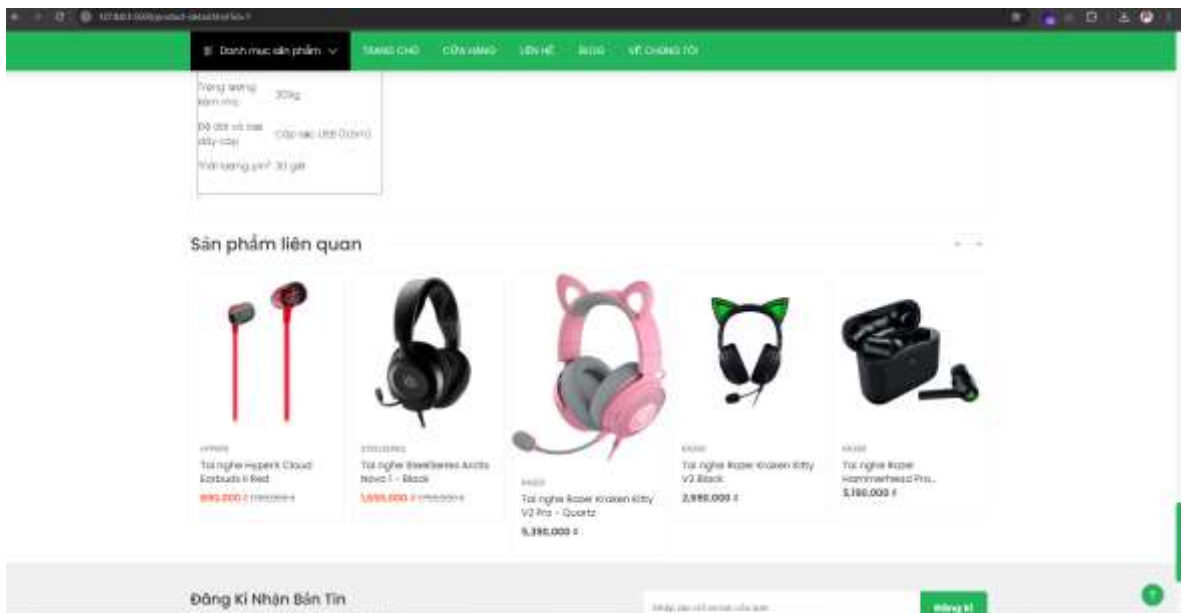
/**
 * Hàm này để tăng số lượng
 */
function plusProduct() {
  const quantity_input = document.getElementById('quantity');
  quantity_input.value = Number(quantity_input.value) + 1;
}

/**
 * Hàm này để giảm số lượng
 */
function minusProduct() {
  const quantity_input = document.getElementById('quantity');
  if (Number(quantity_input.value) > 1)
    quantity_input.value = Number(quantity_input.value) - 1;
}

```

2.4. Chức năng hiển thị sản phẩm liên quan

- Hiển thị sản phẩm liên quan theo danh mục bằng cách sử dụng data file json và get sản phẩm theo danh mục sản phẩm ở trang chi tiết sản phẩm.



Hình 25. Sản phẩm liên quan

```
function initProductDetail() {
  //Fetch dữ liệu từ file json
  fetch('./data/Products.json')
    .then((response) => response.json())
    .then((response) => {
      listProducts = response;
      generateProductByIdQueryParam();
    });
}

/**
 * Hàm này dùng để render sản phẩm liên quan
 * @param listProducts
 * @param idCategory
 * @param idProduct
 */
function generateProductRelateList(listProducts, idCategory, idProduct) {
  const productRelateList = document.querySelector(
    '.product-relate-list .owl-stage-outer .owl-stage'
  );
  if (productRelateList) {
    //Render sản phẩm cùng danh mục và không phải sp chi tiết hiện tại
    let productsRelate = listProducts.filter(
      (item) => item.category == idCategory && item.id != idProduct
    );
    productsRelate.forEach((item) => {
      productRelateList.innerHTML += `<div class="owl-item">
        <div class="product-item">
          <div class="product-thumb">
            <a href="product-detail.html?id=${item.id}">
              
            </a>
          </div>
          <div class="product-info">
            <p>${item.name}</p>
            <p>${item.price}</p>
          </div>
        </div>
      `;
    });
  }
}
```

```

    />
  </a>
</div>
<div class="product-caption">
  <div class="manufacture-product">
    <a href="product-detail.html?id=${item.id}">${item.brand}</a>
  </div>
  <div class="product-name">
    <a href="product-detail.html?id=${item.id}">
      <h4>
        ${item.name}
      </h4>
    </a>
  </div>
  <div class="price-box">
    <span class="regular-price ${
      item.price_old ? 'sale' : ''
    }">${formatVND(item.price)}</span>
    <span class="old-price">${
      item.price_old ? formatVND(item.price_old) : ''
    }</span>
  </div>
  <button class="btn-cart" onclick="addToCart(${
    item.id
  })" type="button">
    Thêm vào giỏ
  </button>
</div>
</div>`;
});
//Owl-carousel
$(document.querySelector('.product-relate-list')).owlCarousel({
  loop: true,
  margin: 10,
  singleItem: true,
  items: 4,
  dots: false,
  nav: true,
  navText: [
    '<i class="bi bi-arrow-left"></i>',
    '<i class="bi bi-arrow-right"></i>',
  ],
  navContainer: '.box-title .custom-product-relate',
  responsive: {
    0: {
      items: 2,
    },
    600: {
      items: 2,
    },
  },
});

```

```

    },
    1000: {
      items: 5,
    },
  },
});
}
}

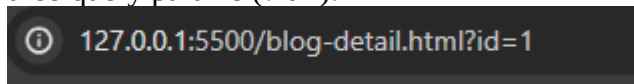
```

Mô tả:

Hàm này được thực hiện cùng với hàm `generateProductByIdParam` để hiển thị những sản phẩm có cùng danh mục sản phẩm

2.5. Chức năng hiển thị chi tiết blog

- Hiển thị chi tiết blog theo từng blog bằng cách sử dụng data file json và get blog theo query params (`?id=`).



Hình 26. Query param blog

```

function initApp() {
  //Fetch dữ liệu từ file json
  const request1 = fetch('./data/Products.json').then((response) =>
    response.json()
  );
  const request2 = fetch('./data/Categories.json').then((response) =>
    response.json()
  );
  const request3 = fetch('./data/Blogs.json').then((response) =>
    response.json()
  );
  Promise.all([request1, request2, request3])
    .then(([data1, data2, data3]) => {
      listProducts = data1;
      listCategories = data2;
      listBlogs = data3;
      generateSearchKey(); //Tạo animation placeholder
      generateCategories(listCategories); // render list danh mục sản phẩm
      generatefeaturedCategoriesList(listCategories); // render danh mục sản phẩm nổi bật
      generateProductList(listProducts); //render list sản phẩm
      generateCollections(listProducts); // render bộ sưu tập sản phẩm
      generateBlogs(listBlogs); //render blogs
      toastMessage();
      loadCartToHTML();
    })
    .catch((error) => {
      console.error(error);
    });
}
/**

```

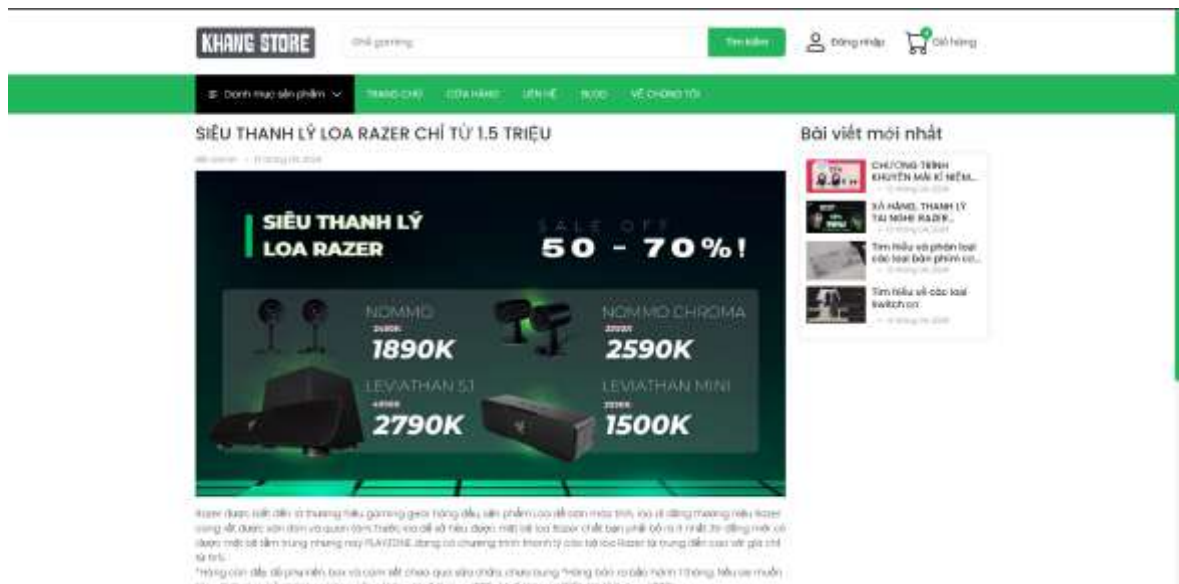
```

* Hàm này dùng để render blogs
* @param listBlogs
*/
function generateBlogs(listBlogs) {
  const blog_list = document.querySelector('.blog-list');
  const blog_main = document.querySelector('.blog-main');
  let blogMain = listBlogs.slice(0, 1)[0]; //Lấy blog mới nhất làm
blog chính
  let blogList = listBlogs.slice(1, 5); // Lấy 4 blog tiếp theo để show
ra giao diện
  //render HTML
  if (blog_list && blog_main) {
    blog_main.innerHTML = `
      <div class="blog-image">
        <a href="blog-detail.html?id=${blogMain.id}">
          
        </a>
      </div>
      <div class="blog-title">
        <h4>
          <a href="blog-detail.html?id=${blogMain.id}">
            ${blogMain.title}</a>
        </h4>
      </div>
      <div class="blog-date">${blogMain.date}</div>
    `;
    blogList.forEach((item) => {
      blog_list.innerHTML += `
        <li class="blog-item">
          <div class="blog-item-image">
            <a href="blog-detail.html?id=${item.id}">
              
            </a>
          </div>
          <div class="blog-item-detail">
            <div class="blog-item-title">
              <a href="blog-detail.html?id=${item.id}"> ${item.title}</a>
            </div>
            <div class="blog-item-date">${item.date}</div>
          </div>
        </li>
      `;
    });
  }
}

```

Mô tả:

Dữ liệu sản phẩm được fetch từ file Blogs.json và render ra giao diện bằng các hàm generateProducts và generateCollections trong đó thẻ a có thuộc href là “product-detail.html?id=\${item.id}” sẽ đưa người dùng đến trang chi tiết sản phẩm có id tương ứng với id của sản phẩm vừa chọn.



Hình 27. Chi tiết blog query params

```
function initBlog() {
  fetch('./data/Blogs.json')
    .then((response) => response.json())
    .then((data) => {
      listBlogs = data;
      generateBlogList(listBlogs);
      generateBlogListLastets(listBlogs);
      generateBlogByIdQueryParams();
    });
}

/**
 * Hàm này dùng để render blog mới nhất
 * @param listBlogs
 */
function generateBlogListLastets(listBlogs) {
  const blog_list = document.querySelector('.blog-list-lastest');
  if (blog_list) {
    const listShow = listBlogs.slice(2, 6); //Lấy 4 blog sau blog đầu tiên
    listShow.forEach((item) => {
      //render html
      blog_list.innerHTML += `
        <li class="blog-item-lastest">
          <div class="blog-image-lastest">
            <a href="blog-detail.html?id=${item.id}">
              
            </a>
          </div>
          <div class="blog-detail-lastest">
            <div class="blog-title-lastest">
              <h3>
                <a href="blog-detail.html?id=${item.id}">${item.title}</a>
              </h3>
            </div>
          </div>
        </li>
      `;
    });
  }
}
```

```

        </div>
        <div class="blog-post-meta">
            <span class="date">${item.date}</span>
        </div>
    </div>
</li>
    `;
    });
}
}

/**
 * Hàm này dùng để render sản phẩm theo id params
 */
function generateBlogByIdQueryParams() {
    const blog_info_detail = document.querySelector('.blog-info-
detail');
    //get params từ url
    const params = new Proxy(new
URLSearchParams(window.location.search), {
        get: (searchParams, prop) => searchParams.get(prop),
    });
    let idBlog = params.id;
    let blog = listBlogs.filter((item) => item.id == idBlog)[0]; //Lọc
ra blog cần tìm
    //render html
    blog_info_detail.innerHTML = `
        <div class="heading-blog my-3"><h3>${blog.title}</h3></div>
        <div class="blog-post-meta">
            <span class="author">Bởi: ${blog.author}</span>
            <span class="date">${blog.date}</span>
        </div>
        <div class="blog-info-image mt-2">
            
        </div>
        <div class="blog-info-content my-3">
            ${blog.content}
        </div>`;
}

```

Mô tả:

Đầu tiên fetch dữ liệu từ file Blogs.json sau đó thực hiện hàm generateBlogByIdParam lấy id query params từ url sau đó thực hiện lọc dữ liệu file Blogs.json trùng với id params và thực hiện render giao diện chi tiết blog

2.6. Chức năng upload ảnh hiển thị ảnh preview

Thêm sản phẩm

Tên sản phẩm:

Giá bán: Giá khuyến mãi: Danh mục sản phẩm:

Mô tả sản phẩm:

Rich Text Editor

↶ ↷ Paragraph

B *I* @ 📷 📅

🗨️ 📺

⌵ ⌵ ⌵ ⌵

Mô tả sản phẩm

Hủy

Lưu

Hình ảnh

⬆

Tải lên

click để tải hình ảnh lên

Hình 28. Trước khi upload ảnh

Thêm sản phẩm

Tên sản phẩm:

Giá bán: Giá khuyến mãi: Danh mục sản phẩm:

Mô tả sản phẩm:

Rich Text Editor

↶ ↷ Paragraph

B *I* @ 📷 📅

🗨️ 📺

⌵ ⌵ ⌵ ⌵

Mô tả sản phẩm

Hủy

Lưu

Hình ảnh

Hình 29. Sau khi upload ảnh

+JS:

```
const btn_upload = document.querySelector('.btn-upload');
const file = document.getElementById('file');
//Nêu có button upload
if(btn_upload)
btn_upload.addEventListener('click', function () {
    file.click();
});

//Nếu có input file upload
if(file)
file.addEventListener('change', updateImageDisplay);
/**
 * Hàm này hiển thị hình ảnh sau khi upload ảnh
 */
function updateImageDisplay() {
    const curFiles = file.files; //File upload
    const imageUpload_icon = document.querySelector('.imageUpload-icon')
    const btn_upload = document.querySelector('.btn-upload')
```

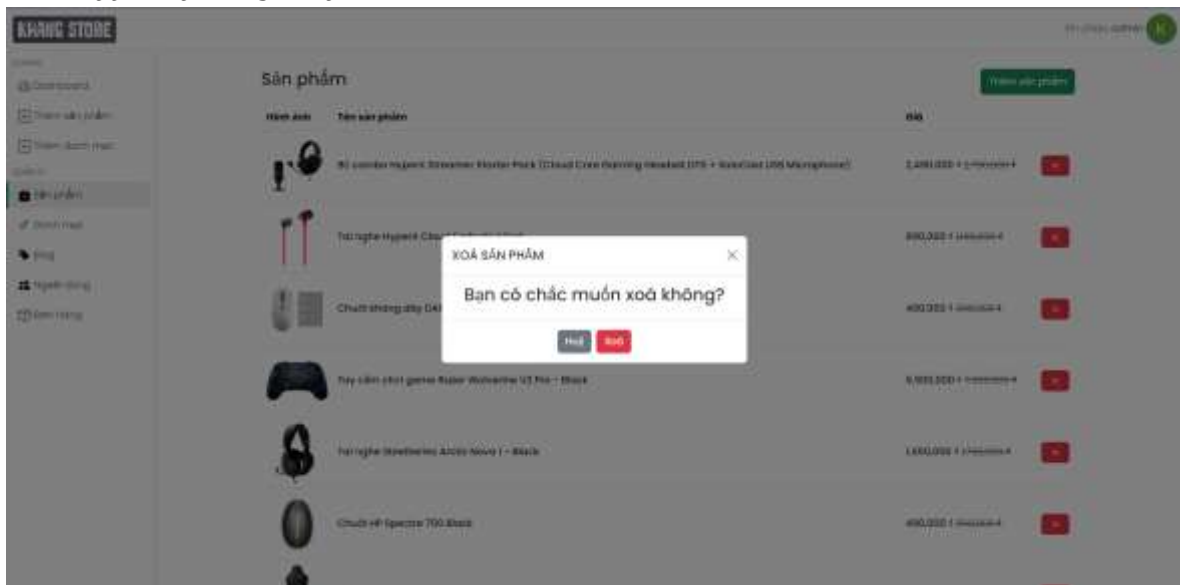
```

const imageUpload_description =
document.querySelector('.imageUpload-description')
image_uploaded.src = URL.createObjectURL(curFiles[0]); //Tạo url
cho file upload và gán vào img
image_uploaded.style.opacity = 1;
//Ẩn nút upload và thông tin upload
imageUpload_icon.style = 'visibility:hidden;'
btn_upload.style = 'visibility:hidden;'
imageUpload_description.style = 'visibility:hidden;'
}
//Nếu có hình upload khi click vào hình thì cho upload hình khác
if(image_uploaded)
image_uploaded.addEventListener('click', function () {
file.click();
});

```

2.7. Chức năng xoá sản phẩm trong dashboard

Khi nhấn vào dấu “X” sẽ xuất ra modal hỏi người dùng có muốn xoá sản phẩm không nếu nhấn nút “Xoá” thì sẽ xoá



Hình 30. Xoá sản phẩm dashboard

+HTML:

```

<!-- Button trigger modal -->
<button
  type="button"
  id="btnModal"
  style="display: none"
  class="btn btn-primary"
  data-bs-toggle="modal"
  data-bs-target="#modal"
>
  Launch demo modal
</button>

<!-- Modal -->
<div

```



```

        class="modal fade"
        id="modal"
        tabindex="-1"
        aria-labelledby="modalLabel"
        aria-hidden="true"
    >
    <div class="modal-dialog modal-dialog-centered">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5" id="modalLabel">XOÁ SẢN
PHẨM</h1>
                <button
                    type="button"
                    class="btn-close"
                    data-bs-dismiss="modal"
                    aria-label="Close"
                ></button>
            </div>
            <div
                class="modal-body w-100 d-flex justify-content-center
align-items-center flex-column"
            >
                <h3>Bạn có chắc muốn xoá không?</h3>
            </div>
            <div class="modal-footer w-100 d-flex justify-content-
center">
                <button
                    type="button"
                    class="btn btn-secondary"
                    data-bs-dismiss="modal"
                >
                    Huỷ
                </button>
                <button
                    type="button"
                    class="btn btn-danger"
                    id="btn-delete"
                    data-bs-dismiss="modal"
                >
                    Xoá
                </button>
            </div>
        </div>
    </div>
</div>

```

+JS:

```

function initDashboardProduct() {
    //Fetch dữ liệu từ file json
    fetch('./data/Products.json')

```

```

        .then((response) => response.json())
        .then((response) => {
            listProducts = response;
            generateProductsDashBoard();
        });
    }
    /**
     * Hàm này render giao diện dashboard danh sách sản phẩm
     */
    function generateProductsDashBoard() {
        const table_body = document.getElementById('table-body');
        table_body.innerHTML = ''
        listProducts.forEach((item) => {
            table_body.innerHTML += `
                <tr>
                    <td>
                        <div class="product-image">
                            <a href="dashboard-edit-
product.html?id=${item.id}">
                                
                            </a>
                        </div>
                    </td>
                    <td>
                        <div class="product-td">
                            <strong
                                ><h6><a href="dashboard-edit-
product.html?id=${
                                item.id
                                }">${item.name}</a></h6></strong>
                            >
                        </div>
                    </td>
                    <td>
                        <div class="product-td">
                            <span style="font-weight:
500;">${formatVND(
                                item.price
                            )}</span>
                            <del class="px-1">${
                                item.price_old ?
formatVND(item.price_old) : ''
                            }</del>
                        </div>
                    </td>
                    <td>

```

```

        <button type="button" class="btn btn-danger"
onclick="confirmDelete({
    item.id
})">
    <i class="bi bi-x"></i>
</button>
</td>
</tr>`;
    });
}

/**
 * Hàm này gọi modal xác nhận trước khi xoá
 * @param id
 */
function confirmDelete(id) {
    const btn_modal = document.getElementById('btnModal');
    btn_modal.click();
    const btn_delete = document.getElementById('btn-delete');
    //Nhấn nút xoá
    btn_delete.onclick = function () {
        listProducts = listProducts.filter((item) => item.id !== id);
        generateProductsDashBoard()
    };
}

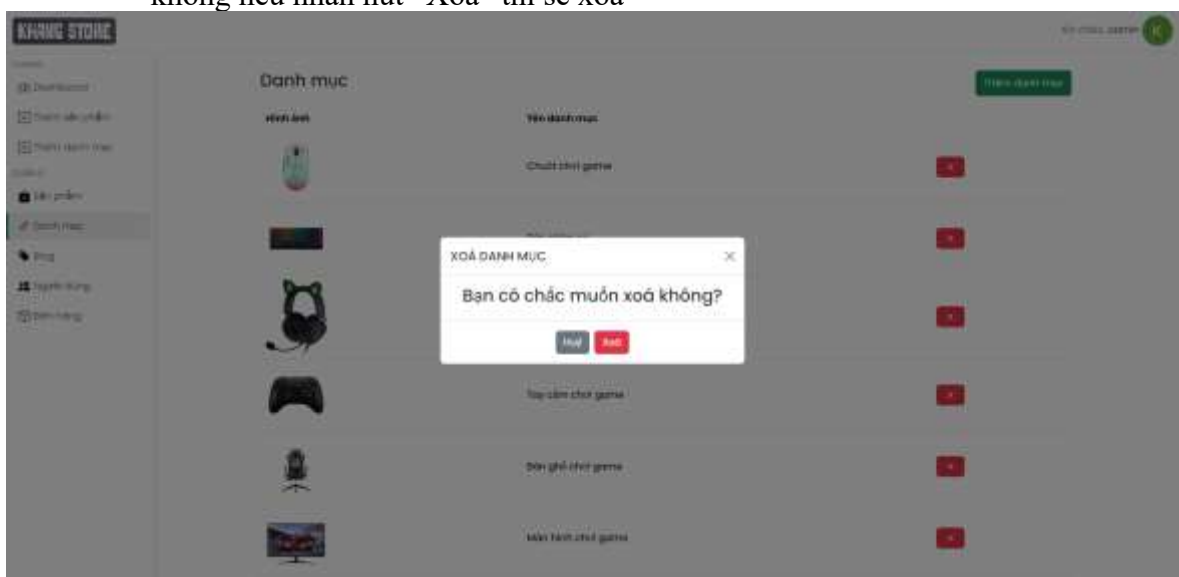
```

Mô tả:

Fetch dữ liệu từ file Products.json và thực hiện render giao diện danh sách dữ liệu cho người dùng. Khi nhấn nút “X” tương ứng với mỗi sản phẩm sẽ xuất hiện modal hỏi người dùng có thật sự muốn xoá sản phẩm hay không. Nếu nhấn nút “xoá” sẽ thực hiện xoá sản phẩm đó ra khỏi mảng danh sách sản phẩm.

2.8. Chức năng xoá danh mục trong dashboard

- Khi nhấn vào dấu “X” sẽ xuất ra modal hỏi người dùng có muốn xoá danh mục không nếu nhấn nút “Xoá” thì sẽ xoá

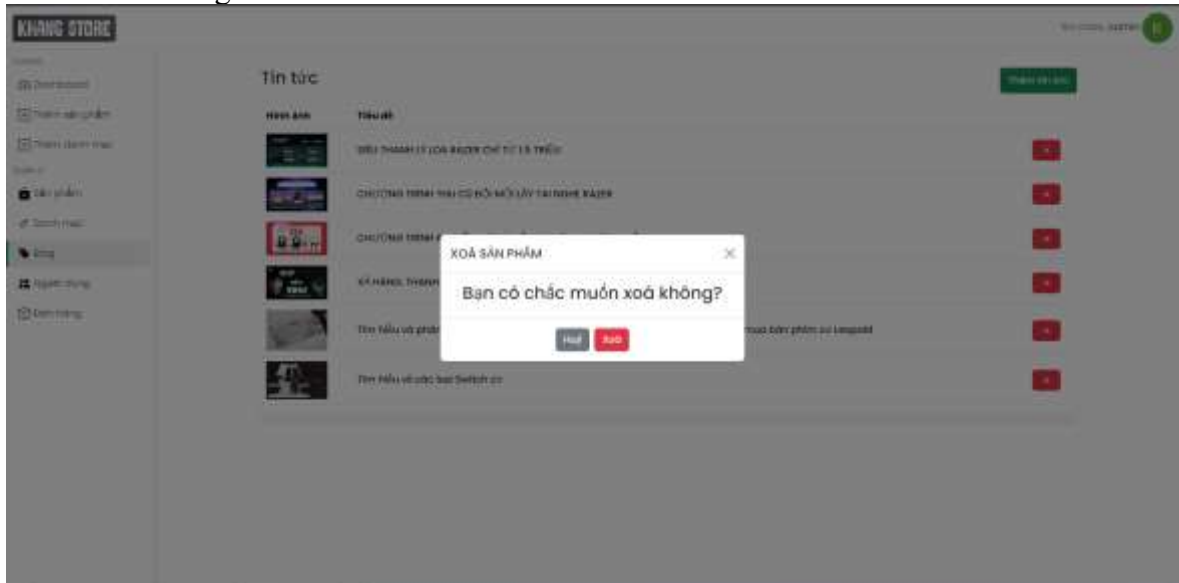


Hình 31. Xoá danh mục trong dashboard

Mô tả code tương tự chức năng xoá sản phẩm trong dashboard (nhưng lấy dữ liệu từ file Categories.json)

2.9. Chức năng xoá blog trong dashboard

- Khi nhấn vào dấu “X” sẽ xuất ra modal hỏi người dùng có muốn xoá sản phẩm không nếu nhấn nút “Xoá” thì sẽ xoá



Hình 32. Xoá blog trong dashboard

Mô tả code tương tự chức năng xoá sản phẩm trong dashboard (lấy dữ liệu từ file Blogs.json)

2.10. Hiển thị số lượng sản phẩm theo danh mục sản phẩm

- Hiển thị số lượng sản phẩm theo danh mục sản phẩm bằng data file json và tính tổng sản phẩm theo danh mục



Hình 33. Số lượng sản phẩm theo danh mục

+JS:

```
function initApp() {  
  //Fetch dữ liệu từ file json  
  const request1 = fetch('./data/Products.json').then((response) =>  
    response.json()  
  );  
  const request2 = fetch('./data/Categories.json').then((response) =>  
    response.json()  
  );  
  const request3 = fetch('./data/Blogs.json').then((response) =>  
    response.json()  
  );  
}
```

```

);
Promise.all([request1, request2, request3])
  .then(([data1, data2, data3]) => {
    listProducts = data1;
    listCategories = data2;
    listBlogs = data3;
    generateSearchKey(); //Tạo animation placeholder
    generateCategories(listCategories); // render list danh mục sản phẩm
    generatefeaturedCategoriesList(listCategories); // render danh mục sản phẩm nổi bật
    generateProductList(listProducts); //render list sản phẩm
    generateCollections(listProducts); // render bộ sưu tập sản phẩm
    generateBlogs(listBlogs); //render blogs
    toastMessage();
    loadCartToHTML();
  })
  .catch((error) => {
    console.error(error);
  });
}

/**
 * Hàm này dùng để render danh mục sản phẩm nổi bật
 * @param listCategories
 */
function generatefeaturedCategoriesList(listCategories) {
  const featured_categories_list = document.querySelector(
    '.featured-categories-list'
  );
  if (featured_categories_list) {
    listCategories.forEach((item) => {
      featured_categories_list.innerHTML += `
        <div class="category-item">
          <div class="category-item-info">
            <h4 class="category-item-name">
              <a href="store.html?idCategory=${item.id}"> ${item.name}
            </a>
          </h4>
          <div class="total-items">${getCountProductsOfCategories(
            item.id
          )} sản phẩm</div>
          <a href="" class="shop-btn">+ Xem thêm</a>
        </div>
        <div class="category-item-thumb">
          <a href="store.html?idCategory=${item.id}">
            
          </a>
        </div>
      `;
    });
  }
}

```

```

        </a>
      </div>
    </div>`;
  });
}
}
}
/**
 * Hàm này để lấy tổng sản phẩm trong một danh mục sản phẩm
 * @param id
 * @returns sum
 */
function getCountProductsOfCategories(id) {
  let sum = 0;
  listProducts.forEach((p) => {
    if (p.category == id) {
      sum++;
    }
  });
  return sum;
}
}

```

2.11. Hiệu ứng animation placeholder tìm kiếm

- Hiệu ứng tự động thay đổi placeholder của input tìm kiếm với 1 mảng các chuỗi cho trước.



Hình 34. Hiệu ứng animation placeholder

+JS:

```

const messages = [
  'Bàn phím akko',
  'Bàn phím keychron',
  'Tai nghe gaming',
  'Ghế gaming',
  'Màn hình',
];
let currentMessageIndex = 0;
let currentCharIndex = 0;
const typingSpeed = 100;
/**
 * Tạo animation placeholder cho input tìm kiếm
 */
function generateSearchKey() {
  if (currentMessageIndex === messages.length) currentMessageIndex = 0;
  // Khi duyệt hết mảng thì quay lại từ đầu

  const message = messages[currentMessageIndex]; // message hiện tại
}

```

```

    if (currentCharIndex < message.length) {
        placeholder += message.charAt(currentCharIndex);
        searchInput.setAttribute('placeholder', placeholder); //generate
từng kí tự
        currentCharIndex++; //kí tự tiếp theo
        setTimeout(generateSearchKey, typingSpeed); //Gọi đệ quy để
generate kí tự tiếp theo
    } else {
        // Sau mỗi message generate xong
        setTimeout(() => {
            currentCharIndex = 0; //đặt lại kí tự đầu tiên
            currentMessageIndex++; //message tiếp theo
            placeholder = ''; //đặt lại placeholder
            searchInput.setAttribute('placeholder', placeholder);
            generateSearchKey(); //Gọi đệ quy
        }, 1000);
    }
}
}

```

2.12. Hiệu ứng zoom highlight

- Hiệu ứng zoom highlight cho ảnh của chi tiết sản phẩm.



Hình 35. Hiệu ứng zoom highlight

```

mediumZoom('.zoom', {
    margin: 50,
    background: '#000',
});

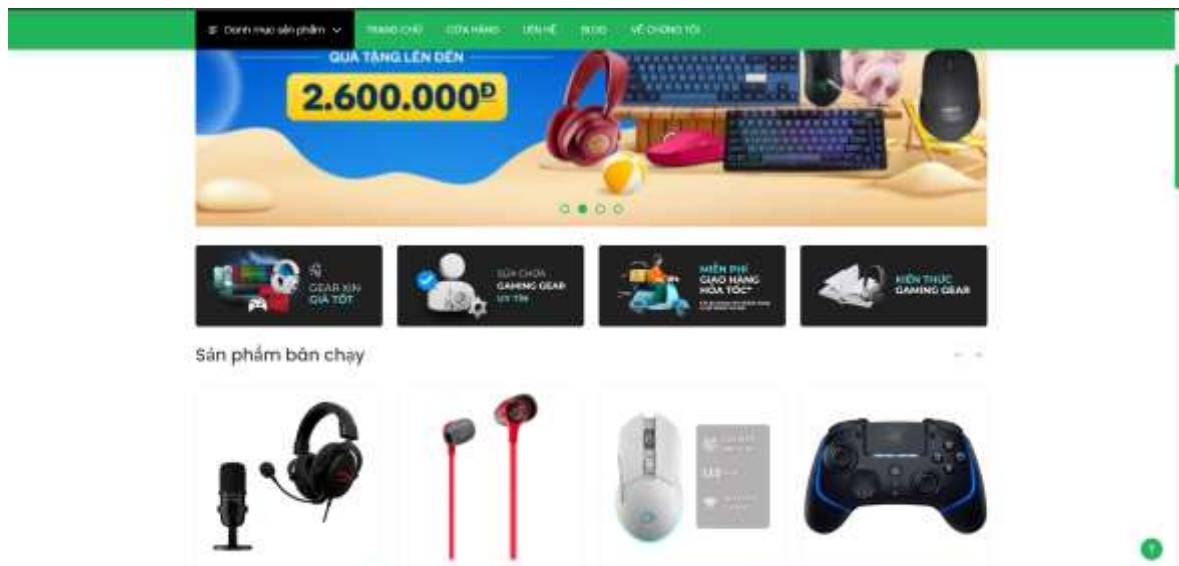
```

Mô tả:

Gọi thư viện mediumZoom thực hiện hiệu ứng zoomhighlight với các img có class là “zoom”

2.13. Hiệu ứng back to top

- Hiệu ứng trở về đầu trang nhấn nút mũi tên để trở về đầu trang



Hình 36. Hiệu ứng back to top

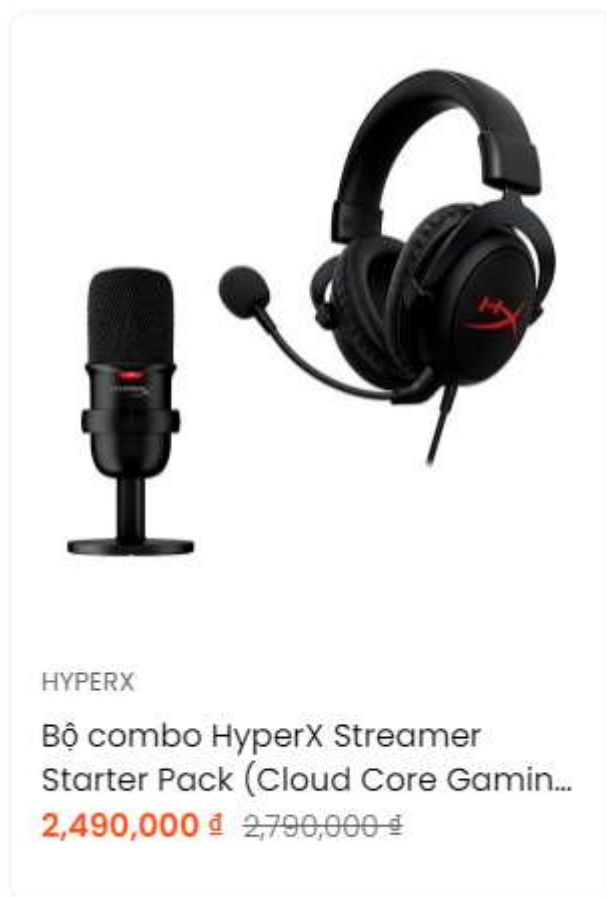
+JS:

```
//Xu ly header top fixed và back to top
window.addEventListener('scroll', () => {
  const currentScroll = window.pageYOffset;
  if (currentScroll > 150) {
    header_bottom.classList.add('sticky');
    back_to_top.classList.add('active');
  } else {
    header_bottom.classList.remove('sticky');
    back_to_top.classList.remove('active');
  }
});
$('.back-to-top').click(function () {
  $('html, body').animate(
    {
      scrollTop: 0,
    },
    800
  );
  return false;
});
```

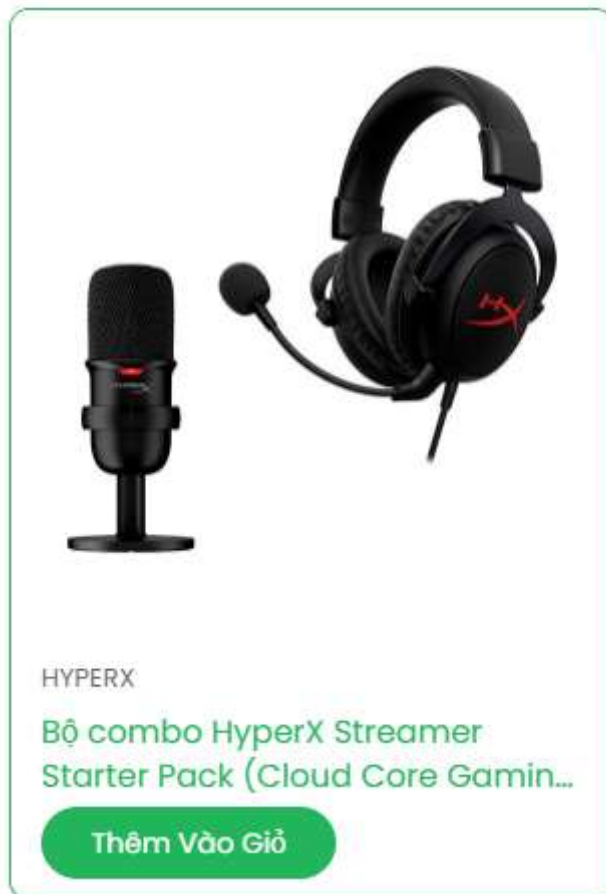
Mô tả:

Khi thanh cuộn xuống lớn hơn 150 thì sẽ hiển thị nút back to top. Khi click nút “.back-to-top” thì sẽ có hiệu ứng cuộn trở về đầu trang

2.14. Hiệu ứng hover sản phẩm



Hình 37. Hiệu ứng trước khi hover sản phẩm



Hình 38. Hiệu ứng sau khi hover sản phẩm
+SCSS:

```
.product-item {
  border: 1px solid $border-color;
  border-radius: 10px;
  transition: 0.3s;
  overflow: hidden;
  height: 100%;
  &:hover {
    border-color: $primary-color;
    .product-caption {
      .price-box {
        transform: translateY(30px);
        opacity: 0;
      }
      .btn-cart {
        opacity: 1;
      }
    }
  }
}

.product-thumb {
  position: relative;
}

.product-caption {
  position: relative;
  padding: 22px 15px 26px;
```

```

.manufacture-product {
  a {
    color: #666;
    font-size: 13px;
    font-weight: 400;
    line-height: 20px;
    text-transform: uppercase;
  }
}

.product-name {
  a {
    white-space: initial;
    overflow: hidden;
    -webkit-line-clamp: 2;
    -webkit-box-orient: vertical;
    display: -webkit-box;
  }
  h4 {
    font-size: 16px;
    font-weight: 400;
    line-height: 22px;
    padding: 5px 0;
    padding-bottom: 0;
    transition: 0.3s;
    &:hover {
      color: $primary-color;
    }
  }
}

.price-box {
  visibility: visible;
  opacity: 1;
  transition: 0.3s;

  .old-price {
    color: #666;
    text-decoration: line-through;
    font-size: 15px;
  }
  .regular-price {
    font-weight: 600;
  }
}

.btn-cart {
  background: $primary-color;
  border: medium none;
  border-radius: 25px;
  color: #ffffff;
  cursor: pointer;
  font-size: 14px;

```

```

line-height: 36px;
margin-top: 10px;
padding: 0 25px;
text-transform: capitalize;
transition: 0.3s;
opacity: 0;
position: absolute;
bottom: 10px;
&:hover {
    background: $secondary-color;
}
}
}
}

```

Mô tả:

Hover sản phẩm đổi màu viền màu chữ và thay thế các thông tin sản phẩm bằng nút “thêm vào giỏ”

2.15. Carousel banner

- Carousel banner tự động chuyển slide, có nút thay đổi slide và nút chọn slide và có thể kéo thả.



Hình 39. Carousel banner

```

$(document).ready(function () {
    $('.banner-carousel').owlCarousel({
        loop: true,
        margin: 0,
        nav: true,
        autoplay: true,
        autoplayTimeout: 5000,
        singleItem: true,
        items: 1,
        navText: [
            '<i class="bi bi-chevron-left"></i>',
            '<i class="bi bi-chevron-right"></i>',
        ],
    });

```

```
});  
});
```

+SCSS:

```
.banner {  
  width: 100%;  
  .owl-dots {  
    position: absolute;  
    bottom: 10px;  
    transform: translateX(-50%);  
    left: 50%;  
    .owl-dot span {  
      width: 15px;  
      height: 15px;  
      background-color: transparent;  
      border: 2px solid $primary-color;  
    }  
    .owl-dot:hover span {  
      background-color: $primary-color;  
    }  
    .owl-dot.active span {  
      background-color: $primary-color;  
    }  
  }  
  .owl-nav {  
    position: absolute;  
    top: 50%;  
    transform: translateY(-50%);  
    width: 100%;  
    .owl-next {  
      position: absolute;  
      right: 50px;  
    }  
    .owl-prev {  
      position: absolute;  
      left: 50px;  
    }  
    .owl-next,  
    .owl-prev {  
      background-color: #ffffff !important;  
      width: 50px;  
      height: 50px;  
      border-radius: 999px;  
      box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.1);  
      cursor: pointer;  
      visibility: hidden;  
      opacity: 0;  
      transition: 0.3s;  
      &:hover {  
        background-color: $primary-color !important;  
        i {  

```

```

        color: #ffffff;
    }
}
i {
    display: flex;
    justify-content: center;
    align-items: center;
    font-size: 20px;
    color: #000000;
}
}
}
}
.owl-carousel:hover {
    .owl-next,
    .owl-prev {
        visibility: visible;
        opacity: 1;
    }
}
}
}

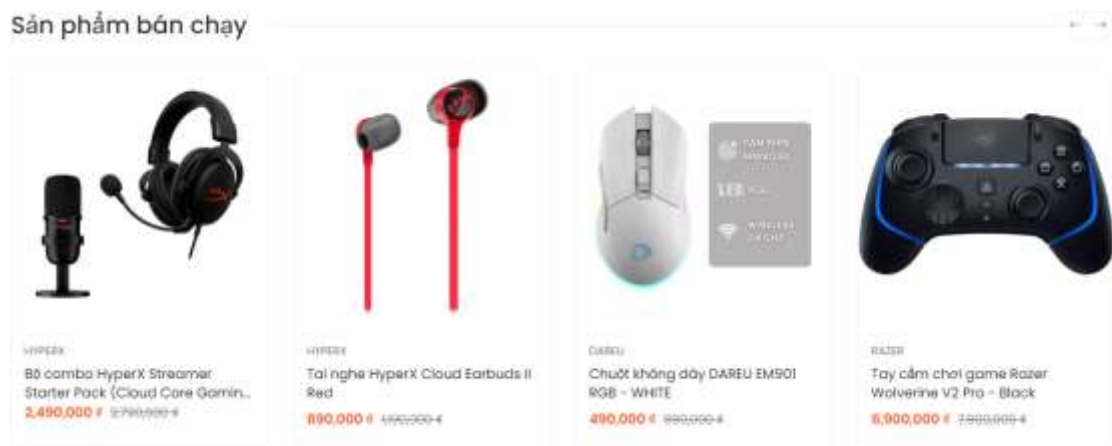
```

Mô tả:

Gọi thư viện owl-carousel set các thuộc tính cho carousel gồm: lặp lại vô hạn, số item trong 1 slide, custom nav, dot, responsive item.

2.16. Carousel sản phẩm

- Carousel sản phẩm có nút thay đổi slide và có thể kéo thả



Hình 40. Carousel sản phẩm

```

$(document.querySelector('.product-list')).owlCarousel({
    loop: true,
    margin: 30,
    singleItem: true,
    items: 4,
    dots: false,
    nav: true,
    navText: [
        '<i class="bi bi-arrow-left"></i>',
        '<i class="bi bi-arrow-right"></i>',
    ],

```

```

    ],
    navContainer: '.box-title .custom-nav-best-seller',
    responsive: {
      0: {
        items: 2,
      },
      600: {
        items: 2,
      },
      1000: {
        items: 4,
      },
    },
  },
});
}

```

+SCSS:

```

.best-seller {
  .custom-nav-best-seller {
    position: absolute;
    top: 0;
    margin-top: 0;
    right: 0;
    background: #fff none repeat scroll 0 0;
    border: 1px solid #f0f0f0;
    border-radius: 25px;
    height: 32px;
    z-index: 100;
    .owl-prev,
    .owl-next {
      border: none;
      padding: 0 8px;
      margin: 0;
      background-color: transparent;
      &:hover {
        background-color: transparent;
      }
      i {
        color: #666;
      }
    }
  }
}

.tab-content {
  margin-top: 20px;
}

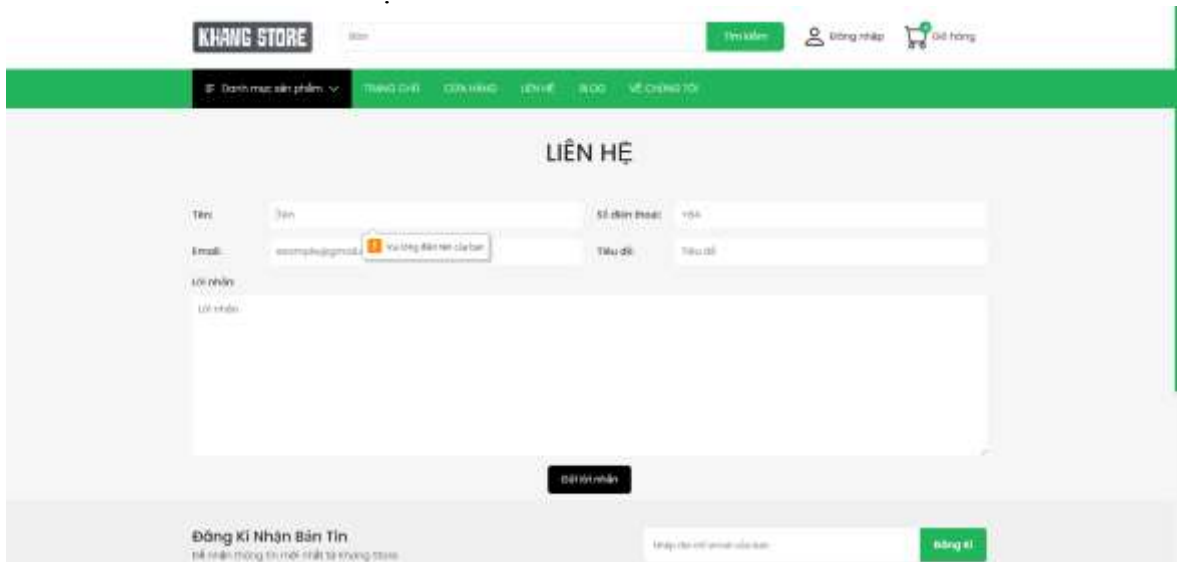
```

Mô tả:

Gọi thư viện owl-carousel set các thuộc tính cho carousel gồm: lặp lại vô hạn, số item trong 1 slide, custom nav, dot, responsive item.

2.17. Validate form

- Validate form liên hệ



Hình 41. Validate form liên hệ

- Khi nhập form liên hệ hợp lệ sẽ xuất ra modal thông báo đã gửi thành công



Hình 42. Gửi thông tin form liên hệ thành công

+HTML:

```
<main>
  <div class="contact">
    <div class="container">
      <div class="row">
        <div class="col-12 text-center py-5">
          <h1>LIÊN HỆ</h1>
        </div>
      </div>
      <div class="row">
        <div class="col-12">
          <div class="contact-message">
            <form class="frm-contact" onsubmit="return false;">
              <div class="row">
                <div
```



```

        class="col-lg-6 col-md-12 col-sm-12 d-flex
align-items-center justify-content-between mb-3"
    >
        <label for="name">Tên: </label>
        <input
            type="text"
            name="name"
            id="name"
            placeholder="Tên"
            required
            oninvalid="setCustomValidity('Vui lòng điền
tên của bạn')"
            oninput="setCustomValidity('')"
        />
    </div>
    <div
        class="col-lg-6 col-md-12 col-sm-12 d-flex
align-items-center justify-content-between mb-3"
    >
        <label for="phone">Số điện thoại: </label>
        <input
            type="text"
            name="phone"
            id="phone"
            placeholder="+84"
            required
            oninvalid="setCustomValidity('Vui lòng điền số
điện thoại của bạn')"
            oninput="setCustomValidity('')"
        />
    </div>
    <div
        class="col-lg-6 col-md-12 col-sm-12 d-flex
align-items-center justify-content-between mb-3"
    >
        <label for="email">Email: </label>
        <input
            type="email"
            name="email"
            id="email"
            placeholder="example@gmail.com"
            required
            oninvalid="setCustomValidity('Vui lòng điền
email của bạn')"
            oninput="setCustomValidity('')"
        />
    </div>
    <div
        class="col-lg-6 col-md-12 col-sm-12 d-flex
align-items-center justify-content-between mb-3"

```

```

>
<label for="subject">Tiêu đề: </label>
<input
  type="text"
  name="subject"
  id="subject"
  placeholder="Tiêu đề"
/>
</div>
<div class="col-12">
  <label for="message">Lời nhắn: </label> <br />
  <textarea
    name="message"
    id="message"
    rows="10"
    placeholder="Lời nhắn"
    minlength="20"
    required
    oninvalid="setCustomValidity('Vui lòng điền
lời nhắn của bạn')"
    oninput="setCustomValidity('')"
  ></textarea>
</div>
<div class="col-12 text-center">
  <button
    class="btn btn-contact"
    id="btn-submit"
    type="submit"
  >
    Gửi lời nhắn
  </button>
</div>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- Button trigger modal -->
<button
  type="button"
  id="btnModal"
  style="display: none"
  class="btn btn-primary"
  data-bs-toggle="modal"
  data-bs-target="#modal"
>
  Launch demo modal
</button>

```

```

<!-- Modal -->
<div
  class="modal fade"
  id="modal"
  tabindex="-1"
  aria-labelledby="modalLabel"
  aria-hidden="true"
>
  <div class="modal-dialog modal-dialog-centered">
    <div class="modal-content">
      <div class="modal-header">
        <h1 class="modal-title fs-5" id="modalLabel">THÀNH
CÔNG</h1>
        <button
          type="button"
          class="btn-close"
          data-bs-dismiss="modal"
          aria-label="Close"
        ></button>
      </div>
      <div class="modal-body w-100 d-flex justify-content-
center align-items-center flex-column">
        <h3>Gửi thông tin thành công</h3></div>
      <div class="modal-footer w-100 d-flex justify-content-
center">
        <button type="button" class="btn btn-success" data-bs-
dismiss="modal">
          OK
        </button>
      </div>
    </div>
  </div>
  <div class="back-to-top"><i class="bi bi-arrow-up"></i></div>
</main>

```

+JS:

```

const form_contact = document.querySelector('.form-contact');
const input_email = document.getElementById('email');
const btn_modal = document.getElementById('btnModal');
const btn_submit = document.getElementById('btn-submit');
btn_submit.addEventListener('click', function () {
  const phone = document.getElementById('phone').value;
  const email = document.getElementById('email').value;
  const name = document.getElementById('name').value;
  const subject = document.getElementById('subject').value;
  const message = document.getElementById('message').value;
  //Form valid thì hiển thị modal thành công và xóa các field đã nhập

```

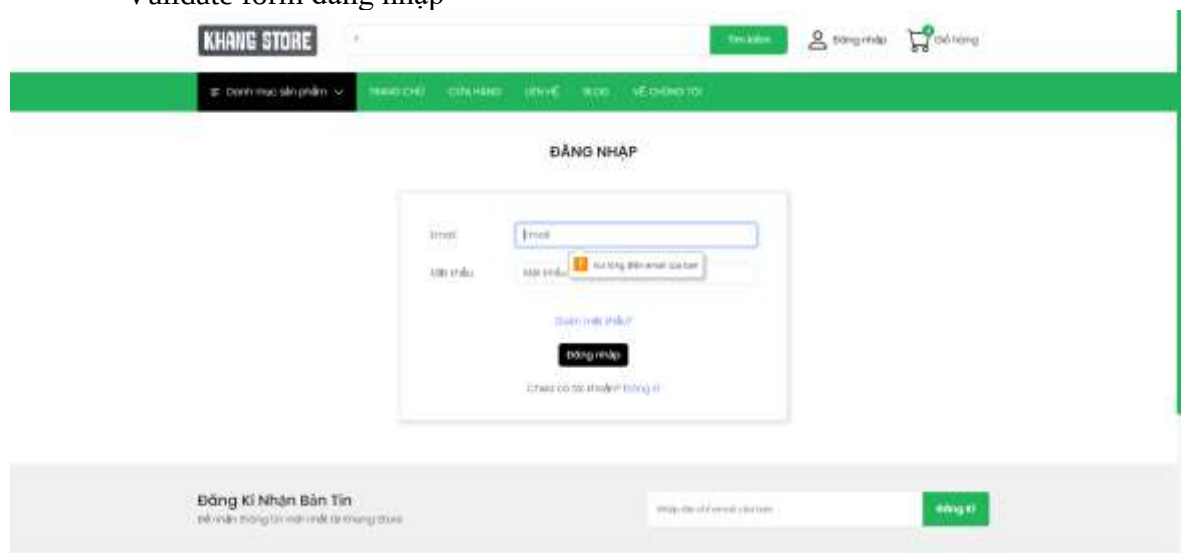
```

    if (name != '' && phone != '' && subject != '' && message != '' &&
validEmail(email)) {
        btn_modal.click();
        document.getElementById('phone').value = ''
        document.getElementById('email').value = ''
        document.getElementById('name').value = ''
        document.getElementById('subject').value = ''
        document.getElementById('message').value = ''
    }
});

/**
 * Hàm này để valid email
 * @param {*} email
 * @returns boolean
 */
function validEmail(email) {
    const regexMatch = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3}$/.test(
        email
    );
    return regexMatch;
}

```

- Validate form đăng nhập



Hình 43. Validate form đăng nhập

- Validate form thêm sản phẩm

Hình 44. Validate form thêm sản phẩm
- Validate form thêm danh mục

Hình 45. Validate form thêm danh mục

2.18. Thư viện đã sử dụng

- Bootstrap 5
- JQuery
- Bootstrap icon
- Owl carousel
- Medium zoom
- ToastJS
- ChartJS
- Ckeditor 5

III. TÀI LIỆU THAM KHẢO

- [1] Tham khảo tài liệu trên: <https://www.w3schools.com/>
- [2] Tham khảo tài liệu trên: <https://www.geeksforgeeks.org/>
- [3] Tham khảo giao diện: <https://themeforest.net/>