

**ASSIGNMENT 2**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

Text

Description automatically generated

TP.HCM, 10 tháng 08 , năm 2023

**GVHD: ThS Mai Thanh Tâm**

**Sinh viên thực hiện:Trần Tiến Đạt – Trần Thiện Nhân**

**MSSV: 2151120028 - 2151120043**

**Lớp HP: 121031**

**MỤC LỤC**

[**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 1**](#_Toc142593835)

[**1. HTML (HyperText Markup Language): 1**](#_Toc142593836)

[**2. CSS (Cascading Style Sheets): 1**](#_Toc142593837)

[**3. JavaScript (JS): 1**](#_Toc142593838)

[**CHƯƠNG 2: TỔNG QUAN CHƯƠNG TRÌNH 2**](#_Toc142593839)

[**1. Phân tích mã lênh của website Bài 1 2**](#_Toc142593840)

[**2. Phân tích mã lệnh của website Bài 2 3**](#_Toc142593841)

[**3. Phân tích mã lệnh của website Bài 3 6**](#_Toc142593842)

[**CHƯƠNG 3: KẾT LUẬN 11**](#_Toc142593843)

[**TÀI LIỆU THAM KHẢO 12**](#_Toc142593844)

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1. **HTML (HyperText Markup Language):**

HTML (Ngôn ngữ Đánh dấu Siêu Văn bản) là ngôn ngữ cơ bản để xây dựng các trang web. Nó sử dụng các thẻ và các phần tử để định dạng và cấu trúc nội dung trên trang web. Với HTML, bạn có thể định nghĩa các phần tử như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng và nhiều phần tử khác. HTML không chỉ định dạng nội dung mà còn xác định cấu trúc của trang web.

1. **CSS (Cascading Style Sheets):**

CSS (Bảng kiểu theo cấp) là một ngôn ngữ định dạng để trình bày và thiết kế giao diện cho các trang web. Nó cho phép bạn thay đổi màu sắc, kích thước, kiểu chữ, định vị và các thuộc tính khác của các phần tử HTML. Với CSS, bạn có thể tạo ra giao diện web đẹp, linh hoạt và dễ dàng bảo trì. CSS cung cấp các lớp kiểu để áp dụng kiểu định sẵn cho các phần tử trên trang web, giúp tách biệt giữa nội dung và kiểu dáng.

1. **JavaScript (JS):**

JavaScript là một ngôn ngữ lập trình phía client mạnh mẽ và đa năng. Nó được sử dụng để tạo ra các tương tác động trên trang web và thực hiện các chức năng phức tạp. JavaScript cho phép bạn điều khiển các phần tử HTML, xử lý sự kiện, thực hiện các phép tính, tạo ra hiệu ứng động, và tương tác với người dùng.

# CHƯƠNG 2: TỔNG QUAN CHƯƠNG TRÌNH

1. **Phân tích mã lênh của website Bài 1**
2. var vitri = 1;
3. var ketthuc = 9;

Biến vitri để lưu trữ vị trí hiện tại của hình ảnh đang được hiển thị trong slideshow. Ban đầu, giá trị của biến này được đặt là 1. Để luôn hiển thị ảnh đầu tiên trước.

Biến ketthuc để lưu trữ tổng số hình ảnh trong slideshow. Trong đây có 9 ảnh nên được đặt là 9, giá trị của biến này được đặt là 9.

Hàm inanh(n):

function inanh(*n*) {

    var image = document.getElementById('slideshow-image');

    var imagePosition = document.getElementById('image-position');

    image.src = '/Assignment/Tainguyen/Anh/' + *n* + '.png';

    imagePosition.innerText = "Ảnh " + *n* + '/' + ketthuc;

}

Đây là hàm để hiển thị hình ảnh tại vị trí n trong slideshow. Trong hàm này, đầu tiên, chúng ta lấy tham chiếu đến phần tử <img> có id là 'slideshow-image' và phần tử có id là 'image-position'. Tiếp theo, chúng ta cập nhật đường dẫn của hình ảnh bằng cách ghép chuỗi 'image/' với số n và phần mở rộng '.png'. Cuối cùng, chúng ta cập nhật nội dung của phần tử 'image-position' để hiển thị vị trí của hình ảnh trong slideshow.

Hàm anhke():

function anhke() {

    vitri++;

    if (vitri > ketthuc) {

        vitri = 1;

    }

    inanh(vitri);

}

Đây là hàm để chuyển đến hình ảnh kế tiếp trong slideshow. Trong hàm này, chúng ta tăng giá trị của biến vitri lên 1. Nếu giá trị mới của vitri vượt quá ketthuc, chúng ta đặt lại giá trị của vitri thành 1. Sau đó, chúng ta gọi hàm inanh(currentImage) để hiển thị hình ảnh .

Hàm anhtruoc():

function anhtruoc() {

    vitri--;

    if (vitri < 1) {

        vitri = ketthuc;

    }

    inanh(vitri);

}

Đây là hàm để chuyển đến hình ảnh trước đó trong slideshow. Trong hàm này, chúng ta giảm giá trị của biến vitri xuống 1. Nếu giá trị mới của vitri nhỏ hơn 1, chúng ta đặt lại giá trị của vitri thành ketthuc. Sau đó, chúng ta gọi hàm inanh(currentImage) để hiển thị hình ảnh mới.

Hàm setinterval()

setInterval(() => {

    anhke();

}, 5000);

Là một hàm có sẳn của hệ thống trong website này được cài đặt để gọi hàm ảnh kế mỗi 5s. với mục đích là cho slideshow tự động chạy mỗi 5s.

1. **Phân tích mã lệnh của website** **Bài 2**

document.querySelector('form[name="frm1"]').addEventListener('submit', function (*event*) {

*event*.preventDefault();

    var isFormValid = kiemTra();

if (isFormValid) {

*this*.submit();

}

});

* Mã lệnh trên là để trình xử lý sự kiện được đính kèm với việc gửi biểu mẫu có tên "frm1". Nó ngăn hành vi gửi biểu mẫu mặc định, xác thực dữ liệu biểu mẫu bằng cách sử dụng hàm có tên kiemTra() và nếu dữ liệu biểu mẫu hợp lệ, nó sẽ tiến hành gửi biểu mẫu.
* **Hàm kiểm tra Kiemtra()**
* function kiemTra() {
* var mssv = document.frm1.mssv.value;
* var ht = document.frm1.ht.value;
* var email = document.frm1.email.value;
* var quocTich = document.frm1.quocTich.value;
* var bosung = document.frm1.bosung.value;
* var gioiTinh = document.frm1.gender.value;
* var checkboxes = document.querySelectorAll('input[name="hobby"]:checked');
* var soThich = *Array*.from(checkboxes).map(function(*checkbox*) {
* return *checkbox*.value;
* });
* var loi = "";
* if (mssv.length == 0) {
* loi = loi + "MSSV không được để trống <br>";
* document.frm1.mssv.classList.add("bg-warning");
* }
* if (ht.length <= 0 || ht.length > 30) {
* loi = loi + "Họ tên không được bỏ trống <br>";
* document.frm1.ht.classList.add("bg-warning");
* }
* var s = parseInt(mssv);
* if (email == '') {
* loi = loi + "Email không được để trống<br>";
* document.frm1.email.classList.add("bg-warning");
* }
* if (gioiTinh == '') {
* loi = loi + "Bạn chưa chọn giới tính<br>";
* document.getElementById("gender\_input").classList.add("bg-warning");
* }
* if (soThich.length == 0) {
* loi = loi + "Bạn chưa chọn sở thích";
* document.getElementById("hobbies\_input").classList.add("bg-warning");
* }
* if (quocTich == 0) {
* loi = loi + "Bạn chưa chọn quốc tịch<br>";
* document.frm1.quocTich.classList.add("bg-warning");
* }
* if (bosung.length < 50 || bosung.length > 200) {
* loi = loi + "Hãy nhập Thông tin bổ sung ít hơn 200 ký tự";
* document.frm1.bosung.classList.add("bg-warning");
* }
* if (loi.length > 0) {
* document.getElementById("loi").innerHTML = loi;
* }
* return false;
* }
* **Khai báo các biến cần thiết:**

**+Lần lượt khai báo các biến: mssv, ht, email, quoctich, bosung, gioi tinh. Để nhận giá trị lần lượt là mã số sinh viên, họ tên, email, quốc tịch, bổ sung, giới tính.**

**+khai báo một checkboxes: để tạo một bảng chọn gồm các sở thích.**

**+khai báo lỗi : để lưu lại các lỗi cần sửa đổi.  
- khiểm tra**

**+ khiểm tra mã số sinh viên: nếu chiều đài của mssv = 0 đồng nghĩa với xâu trống thì lưu vào biến loi “MSSV không được để trống” và lưu lại cảnh báo.**

**+ khiểm tra họ tên: để họ tên không trống và không quá 30 kí tự. Nếu vượt quá thì bị vi phảm lỗi và lưu vào biên loi “ Họ Tên không được để trống” và lưu lại cảnh báo.**

**+ khiểm tra email: nếu chiều dài xâu email trống thì bị phạm lỗi và lưu vào biên loi “email không được để trống” và lưu lại cảnh báo.**

**+ khiểm tra giới tính : nếu giới tính trống ta biết được chưa chọn giới tính nào lưu vào biên loi “ Bạn chưa chọn giới tính nào” và lưu lại cảnh báo.**

**+ khiểm tra sở thích : nếu sở thích trống ta biết được chưa chọn sở thích nào lưu vào biên loi “ Bạn chưa chọn sở thích nào” và lưu lại cảnh báo.**

**+ khiểm tra quốc tịch : nếu quốc tịch trống ta biết được chưa chọn quốc tịch nào lưu vào biên loi “ Bạn chưa chọn quốc tịch nào” và lưu lại cảnh báo.**

**+ khiểm tra thông tin bổ sung: nếu thông tin bổ sung lớn hơn 200 ký tự ta lưu vào biên loi “ Hãy nhập thông tin bổ sung ít hơn 200 ky tự “ và lưu lại cảnh cáo.**

**+ khiểm tra biến loi: nêu length.loi>0 đồng nghĩa với đã có lỗi xãy ra ta xuất lỗi .**

**-Cuối cùng, hàm trả về false, đây là một phương pháp phổ biến khi xử lý xác thực biểu mẫu bằng JavaScript. Điều này có thể ngăn không cho biểu mẫu được gửi nếu có lỗi xác thực.**

1. **Phân tích mã lệnh của website Bài 3.**

* Hàm filter(selectElement):

 function filter(*selectElement*) {

     var selectedValue = *selectElement*.value;

     var minPrice, maxPrice;

     var tbody = document.querySelector('tbody#listProducts');

     var product\_rows = tbody.querySelectorAll('tr');

     // Set the minPrice and maxPrice based on the selected value

     switch (selectedValue) {

         case '0':

             minPrice = 0;

             maxPrice = Infinity;

             break;

         case '1':

             minPrice = 0;

             maxPrice = 100;

             break;

         case '2':

             minPrice = 100;

             maxPrice = 200;

             break;

         case '3':

             minPrice = 200;

             maxPrice = 500;

             break;

         case '4':

             minPrice = 500;

             maxPrice = Infinity;

             break;

         default:

             // No valid selection, do nothing

             return;

     }

     for (var i = 0; i < product\_rows.length - 1; i++) {

         var priceProducts = product\_rows[i].cells[2].innerText;

         if (priceProducts < minPrice || priceProducts > maxPrice) {

             product\_rows[i].style.display = 'none';

         } else {

             product\_rows[i].style.display = 'table-row';

         }

     }

 }

**Hảm trên dùng để lọc khoảng giá :**

* **var selectValue = selectElement.value;:** Dòng này nhận giá trị của tùy chọn đã chọn trong menu thả xuống.
* **var minPrice, maxPrice;:** Các biến này sẽ được sử dụng để lưu phạm vi giá tối thiểu và tối đa để lọc.
* **var tbody = document.querySelector('tbody#listProducts');:** Dòng này chọn phần tử <tbody> có ID là "listProducts" từ tài liệu HTML. Giả định rằng nội dung bảng này chứa các hàng bạn muốn lọc.
* **var product\_rows = tbody.querySelectorAll('tr');:** Dòng này chọn tất cả các hàng trong nội dung bảng và lưu trữ chúng trong biến product\_rows.

- Câu lệnh switch được sử dụng để xác định các giá trị minPrice và maxPrice phù hợp dựa trên tùy chọn đã chọn. Mỗi trường hợp tương ứng với một khoảng giá khác nhau.

- Vòng lặp for lặp qua từng hàng trong mảng product\_rows (ngoại trừ hàng cuối cùng) và thực hiện các bước kiểm tra sau: Nó truy xuất giá của sản phẩm từ ô thứ ba của hàng hiện tại bằng cách sử dụng product\_rows[i].cells[2].innerText. Nếu giá nằm ngoài phạm vi đã chỉ định (giá tối thiểu đến giá tối đa), hàng sẽ bị ẩn bằng cách đặt thuộc tính kiểu hiển thị của nó thành 'không'. Nếu giá nằm trong phạm vi, hàng sẽ được hiển thị bằng cách đặt thuộc tính kiểu hiển thị của nó thành 'table-row'. Về cơ bản, mã sẽ lặp qua các hàng trong bảng, ẩn các hàng có giá không khớp với phạm vi đã chọn và hiển thị các hàng khớp với phạm vi.

* Hàm calculateTotal(obj)

 function calculateTotal(*obj*) {

     var quantity = *obj*.value;

     var row = *obj*.parentNode.parentNode;

     var price = row.cells[2].innerText;

     if (quantity < 0) {

         quantity = 0;

*obj*.value = 0;

     }

     var totalPrice = parseInt(quantity) \* parseInt(price);

     if (*Number*.isFinite(totalPrice) == false) {

         totalPrice = '';

     }

     row.cells[4].innerText = totalPrice;

     updateCart();

 }

* **Var quantity = obj.value :** Dòng này nhận giá trị (số lượng) do người dùng nhập vào, thường là từ một trường đầu vào được liên kết với hàng.
* **Var row = obj.parentNode.parentNode;** Dòng này tìm thấy hàng cha (<tr>) của phần tử đầu vào đã kích hoạt phép tính. Phần tử đầu vào thường được tìm thấy trong một ô (<td>) trong một hàng của bảng.
* **var price = row.cells[2].innerText;** Dòng này trích xuất giá từ ô thứ ba của hàng (giả sử lập chỉ mục dựa trên số không). Điều này giả định rằng giá được lưu trữ dưới dạng nội dung văn bản trong ô đó.
* Sau đó, mã sẽ kiểm tra xem số lượng đã nhập có nhỏ hơn 0 hay không. Nếu đúng như vậy, mã sẽ đặt số lượng thành 0 và cập nhật giá trị đầu vào thành 0 để ngăn số lượng âm.
* Tổng giá được tính bằng cách nhân số lượng và giá được phân tích cú pháp. Phép nhân sử dụng parseInt để đảm bảo số lượng và giá được coi là số nguyên.
* Sau đó, mã sẽ kiểm tra xem tổng Giá được tính có phải là một số hữu hạn hay không bằng cách sử dụng Number.isFinite(). Nếu đó không phải là một số hữu hạn (điều này có thể xảy ra nếu số lượng hoặc giá không phải là một số hợp lệ), thì giá tổng được đặt thành một chuỗi trống.
* row.cells[4].innerText = totalPrice;: Dòng này đặt nội dung của ô thứ năm (giả sử lập chỉ mục dựa trên số 0) trong cùng một hàng thành giá tổng được tính. Ô này là nơi hiển thị tổng giá cho hàng đó.
* updateCart();: Dòng này có lẽ gọi một hàm có tên updateCart() để cập nhật tổng số giỏ hàng hoặc thực hiện bất kỳ phép tính cần thiết nào khác.
* Hàm updateCart()

 function updateCart() {

     var cartTable = document.getElementById('cart');

     cartTable.innerHTML = '';

     var selectedProducts = document.querySelectorAll('.chon:checked');

     for (var i = 0; i < selectedProducts.length; i++) {

         var productRow = selectedProducts[i].parentNode.parentNode;

         var productName = productRow.cells[1].innerText;

         var productPrice = productRow.cells[2].innerText;

         var productQuantity = productRow.querySelector('.quantity').value;

         /\*if (productQuantity < 0) {

             productQuantity = 0;

         }\*/

         var productTotal = productRow.cells[4].innerText;

         var cartRow = cartTable.insertRow();

         var cartNameCell = cartRow.insertCell(0);

         var cartPriceCell = cartRow.insertCell(1);

         var cartQuantityCell = cartRow.insertCell(2);

         var cartTotalCell = cartRow.insertCell(3);

         cartNameCell.innerText = productName;

         cartPriceCell.innerText = productPrice;

         cartQuantityCell.innerText = productQuantity;

         cartTotalCell.innerText = productTotal;

     }

     calculateCartTotal();

 }

* **var cartTable = document.getElementById('cart');** **cartTable.innerHTML = '';:** Phần này xóa nội dung của bảng giỏ hàng có ID "giỏ hàng" bằng cách đặt HTML bên trong của nó thành một chuỗi trống. Điều này chuẩn bị bảng giỏ hàng để cập nhật các sản phẩm đã chọn.
* **var selectProducts = document.querySelectorAll('.chon:checked');:** Dòng này chọn tất cả các hộp kiểm đã chọn có lớp "chon". Điều này có thể đại diện cho các hộp kiểm được liên kết với các sản phẩm đã chọn trong danh sách sản phẩm chính.

Sau đó, mã đi vào một vòng lặp lặp qua từng sản phẩm đã chọn:

* Nó truy xuất tên, giá và số lượng của sản phẩm từ các ô tương ứng trong hàng sản phẩm.
* Nó tạo một hàng mới trong bảng giỏ hàng và chèn các ô để hiển thị tên sản phẩm, giá, số lượng và tổng.
* Nó đặt nội dung của các ô này với các giá trị tương ứng.
* Tính toánCartTotal();: Dòng này có lẽ gọi một hàm có tên là tính toánCartTotal() để cập nhật tổng chi phí của các mặt hàng trong giỏ hàng.
* Hàm calculateCartTotal(obj)
* function calculateCartTotal(*obj*) {
* var cartTable = document.getElementById('cart');
* var cartRows = cartTable.rows;
* var cartTotal = 0;
* for (var i = 0; i < cartRows.length; i++) {
* var rowTotal = parseFloat(cartRows[i].cells[3].innerText);
* if (*Number*.isFinite(rowTotal) == false) {
* rowTotal = 0;
* }
* cartTotal = cartTotal + rowTotal;
* }
* document.getElementById("tong").innerHTML = cartTotal;
* }
* var cartTable = document.getElementById('cart');: Dòng này lấy phần tử bảng giỏ hàng có ID là "giỏ hàng".
* var cartRows = cartTable.rows;: Dòng này truy xuất một tập hợp các hàng dạng mảng trong bảng giỏ hàng.
* var cartTotal = 0;: Biến này sẽ được sử dụng để lưu trữ tổng chi phí tích lũy của các mặt hàng trong giỏ hàng. Mã đi vào một vòng lặp lặp qua từng hàng trong bảng giỏ hàng: Nó truy xuất tổng giá trị chi phí từ ô thứ tư (giả sử lập chỉ mục dựa trên số 0) của hàng hiện tại. Nó sử dụng parseFloat để chuyển đổi tổng giá trị chi phí thành một số dấu phẩy động. Nếu giá trị được chuyển đổi không phải là một số hữu hạn (có thể xảy ra nếu tổng giá trị chi phí không hợp lệ), giá trị này sẽ đặt rowTotal thành 0.
* Nó thêm rowTotal vào cartTotal để tích lũy tổng chi phí của tất cả các mặt hàng. document.getElementById("tong").innerHTML = cartTotal;: Dòng này cập nhật một phần tử HTML có ID "tong" (có thể dùng để hiển thị tổng số giỏ hàng) với tổng số giỏ hàng đã tính.
* Hàm reverAttribute(obj)
* function reverseAttribute(*obj*) {
* // đảo trạng thái khi tick vào checkbox
* var node = *obj*.parentElement.parentElement;
* var input = node.getElementsByTagName("input")[1];
* if (input.hasAttribute("readonly")) {
* input.removeAttribute("readonly");
* input.style.opacity = 1;
* } else {
* input.setAttribute("readonly", "readonly");
* input.style.opacity = 0.5;
* input.value = 0;
* }
* calculateTotal(*obj*)
* }
* **var node = obj.parentElement.parentElement**;: Dòng này lấy phần tử cha của phần tử cha của hộp kiểm. Giả định là hộp kiểm nằm trong một ô trong một hàng (cấu trúc bảng).
* **var input = node.getElementsByTagName("input")[1];**Dòng này tìm phần tử đầu vào trong cùng một hàng bằng cách sử dụng phương thức getElementsByTagName và chỉ mục [1] (giả sử lập chỉ mục dựa trên số 0). Điều này dựa trên giả định rằng có hai phần tử đầu vào trong mỗi hàng và phần tử thứ hai là mục tiêu để chuyển đổi thuộc tính chỉ đọc.

Sau đó, mã sẽ kiểm tra xem phần tử đầu vào có thuộc tính chỉ đọc hay không:

* Nếu nó có thuộc tính chỉ đọc, mã sẽ xóa thuộc tính chỉ đọc, đặt độ mờ của đầu vào thành 1 (hiển thị đầy đủ) và giữ nguyên giá trị đầu vào.
* Nếu nó không có thuộc tính chỉ đọc, mã sẽ đặt thuộc tính chỉ đọc thành "chỉ đọc", giảm độ mờ của đầu vào xuống 0,5 (hiển thị một nửa) và đặt giá trị đầu vào thành 0.
* Cuối cùng, phép tínhTotal(obj) được gọi, có lẽ là để cập nhật phép tính tổng dựa trên những thay đổi được thực hiện đối với trạng thái của phần tử đầu vào.

# CHƯƠNG 3: KẾT LUẬN

Trong bài code web về swift snack, đã được thực hiện nhiều chức năng quan trọng để tạo ra một trang web hoạt động mượt mà và hấp dẫn cho người dùng. Các chức năng quan trọng bao gồm:

Danh mục sản phẩm: Trang web có sắp xếp sản phẩm vào các danh mục rõ ràng, giúp người dùng dễ dàng tìm kiếm và duyệt qua các sản phẩm một cách thuận tiện. Mỗi sản phẩm có hình ảnh, mô tả và giá cả, giúp khách hàng có đủ thông tin để quyết định mua hàng.

Chức năng mua hàng: Khách hàng có thể nhập số lượng sản phẩm cần mua và xem tổng giá trị đơn hàng trước khi thanh toán.

Hơn nữa, khách hàng còn có thể đăng ký trở thành thành viên mua hàng của website để có nhiều ưu đãi trong quá trình mua sắm sau này.

Những chức năng trên đã góp phần tạo ra một trang web cơ bản có thể đáp ứng nhu cầu mua sắm snack trực tuyến của người dùng. Tuy nhiên, do thời gian còn hạn chế những chức năng trên còn chưa có sự đột phá. Trong tương lai, chúng em sẽ cố gắng tiếp tục phát triển để những chức năng này hoàn thiện hơn.

# 

# TÀI LIỆU THAM KHẢO

<WEB1042 - Assignment > Ngày truy cập (03/08/2023)

<<https://freetuts.net/hoc-javascript/javascript-can-ban>> Ngày truy cập (03/08/2023)

<<https://chat.openai.com/>> Ngày truy cập (03/08/2023)