**Graphical user interface, text

Description automatically generated**

**Báo Cáo Lập Trình Cơ Sở Vơi Javascript**

**(WEB1043)**

**GVHD:** **Thầy Nguyễn Văn Long**

**Họ và tên: Đỗ Gia Huy**

**MSSV: PS26598**

**Lớp : IT18203**

**TPHCM - 2022**

1. Trang danh sách sản phẩm listsp.html

Bước 1:

Text

Description automatically generated

Tạo ra mảng listsp trong mỗi phần tử của mảng chứa dữ đối tượng bao gồm : Tên sản phẩm, giá, hình.

Bước 2:

Text

Description automatically generated

* Trong phần body tạo ra 2 thẻ header và section.
* Trong phần header tạo ra thẻ <b id="soluong"></b> để hiển thị số lượng sản phầm được thêm.
* Trong phần section cho vòng lặp for lặp qua các phần tử của listsp, sau đó lấy ra tên sản phẩm, giá sản phẩm và hình của từng phần tử trong mảng.
* Dùng câu lệnh document.write để hiển thị ra.
* Tạo ra một thẻ div bao bọc trong document.write có 2 hàm onmouseover="hienicon(this)" onmouseout="anicon(this)". Dùng để ẩn hiện icon khi thao tác chuột ra vào sản phẩm

Text

Description automatically generated

(Cho chọn tới node có id là icons để chỉnh sửa thuộc tính display).

* Thẻ img để cho hiển thị hình sản phẩm và thẻ div có class info để thể hiện giá và tên sản phẩm

Text

Description automatically generated

* Thẻ div có class icons để chứa các icon

Text

Description automatically generated

+ Thẻ img đầu tiên có chứa hàm xemchitiet(), khi sản phẩm đó được ấn thì sẽ thông báo và chuyển qua trang chi tiết sản phẩm bằng location

Text

Description automatically generated

+ Thẻ img thứ 2 chứa hàm chonsp(), cho số lượng sản phẩm ban đầu bằng 0 sau đó khi ấn vào sản phẩm bất kì thì số lượng sẽ tăng lên và hiện thị ra.

Text

Description automatically generated

+Thẻ img thứ 3 chứa hàm thich(), khi chọn vào icon thích thì sẽ đổi sang icon là đã thích và ngược lại.

Text

Description automatically generated

Bước 3: Css cho đẹp

<style>

        \* {

            margin: 0;

            box-sizing: border-box;

        }

*#sanphammoi* {

            width: 1000px;

            margin: auto;

            display: grid;

            grid-template-columns: 32.5% 32.5% 33%;

            grid-template-rows: 400px;

            grid-column-gap: 1%;

            grid-row-gap: 10px;

        }

*#sanphammoi* *.sp* {

            border: 1px solid navy;

            text-align: center;

            border-radius: 5px;

            position: relative;

        }

*#sanphammoi* *.hinhsp* {

            height: 300px;

            max-width: 100%;

        }

*#sanphammoi* *.icons* {

            position: absolute;

            top: 140px;

            left: 120px;

            background-color: lightblue;

            display: none;

        }

*#sanphammoi* *.gia* {

            font-size: 1.3em;

            padding: 10px 0px;

        }

*#sanphammoi* *.tensp* {

            padding: 0px 25px;

            height: 40px;

        }

        header{

            width: 1000px;

            height: 120px;

            margin: auto;

            background-color: aqua;

            position: relative;

        }

        header *#soluong*{

            position: absolute;

            bottom: 10px;

            right: 10px;

            color: rebeccapurple;

        }

    </style>

Kết quả:

Graphical user interface, website

Description automatically generated

2. Trang slide show  slider.html

Bước 1:

Text

Description automatically generated

Tạo ra hai mảng arr\_hinh để chứa các hình và mảng arr\_mota để chứa các mô tả.

Chèn đường link sau trong thẻ head



Bước 2:

Text

Description automatically generated

Trong body tạo ra thẻ div có id=’’slide’’ chứa thẻ img để hiện thị hình ảnh, thẻ b để hiển thị mô tả và các nút prev, next

Trong các nút prev và next tạo prev() và next() với sự kiện oncick

Bước 3:

Text

Description automatically generated

* Tạo thẻ script ở cuối trang.
* Tạo biến index = 0;
* Trong hàm prev cho biến index giảm nếu biến index nhỏ hơn 0 thì cho index bằng độ dài mảng arr\_hinh – 1.
* Trong hàm next cho biến index tăng nếu biến index lớn hơn arr\_hinh.length thì cho index bằng 0.
* Trong cả hai hàm trên đều trỏ tới node có id là hinh để hiển thị hình bằng phần tử của mảng arr\_hinh qua giá trị index: arr\_hinh[index].
* Tương tự mô tả cũng được thể hiện qua: arr\_mota[index].
* Tạo hàm setInterval trong đó gọi hàm next đã tạo ở trên, sau 5s thì sẽ thực hiện hàm next và hình sẽ thay đổi.

Bước 4: Css cho đẹp

<style>

*#slide* {

            width: 1000px;

            height: 350px;

            margin: auto;

            position: relative;

        }

*#slide* *#hinh* {

            width: 100%;

            height: 100%;

        }

*#slide* *#mota* {

            position: absolute;

            top: 90%;

            left: 40%;

            color: red;

            font-size: 1.2em;

        }

*#slide* *.fa-chevron-circle-left* {

            position: absolute;

            top: 45%;

            left: 30px;

            font-size: 32px;

            color: red;

        }

*#slide* *.fa-chevron-circle-right* {

            position: absolute;

            top: 45%;

            right: 30px;

            font-size: 32px;

            color: red;

        }

    </style>

Kết quả:   
Graphical user interface, website

Description automatically generated

3. Trang giỏ hàng giohang.html

Bước 1:

Text

Description automatically generated

Trong thẻ body tạo div chứa 3 button Tạo cart, hiện cart và xóa cart. Sau đó tạo 1 thẻ div để hiện thị Giỏ hàng và 1 div để hiển thị tổng tiền giỏ hàng

Bước 2:

Graphical user interface, text

Description automatically generated

* Trong hàm taoCart() tạo biến cart là mảng chứa các phần tử có mỗi đối tượng gồm tensp, gia, hinh, soluong.
* Câu lệnh sessionStorage.setItem() để tạo lưu cart theo kiểu chuỗi json nhờ hàm JSON.stringify().

Text

Description automatically generated

* Trong hàm hienCart() tạo biến str bằng thẻ div để hiển thị .
* Biến text = giá trị lấy ra từ sessionStorage đã tạo trước đó.
* Biến cart để chuyển giá trị của biến text từ chuỗi Json thành object trong javascipt để xử lí thông qua lệnh Json.parse()
* Cho vòng lặp for lặp qua tất cả các phần tử trong mảng để hiển thị giá trị.

Trong tag input cho hai sự kiện là onchange="doisoluong()" onkeyup="doisoluong()"

Text

Description automatically generated

* Khi thay đổi số lượng trong tag input thì hàm doisoluong() sẽ chạy.
* Cho biến soluong = giá trị trong ô input.
* Biến gia được chọn tới bằng cách chọn tới node cha của thẻ input rồi lấy node trước của node soluong.
* Biến tien = soluong\*gia
* Chọn tới node sau của node hiển thị số lượng(bằng cách chọn tới node cha tag input rồi chọn node kế tiếp của thẻ cha vừa chọn) rồi cho hiển thị tiền vừa tính.
* Sau đó hàm tongtiengiohang() được thực hiện.

Text

Description automatically generated

* Trong hàm tongtiengiohang() cho biến arr chứa mảng được lấy bởi className(“row”).
* Biến tongtienGH = 0;
* Cho vòng lặp for chạy qua mảng arr.

+ Trong vòng lặp cho biến row = phần tử đang xét.

+ Biến tien = phần tử đang xét trỏ tới phần tử tagname(“span”)[0] để lấy giá trị sau đó chuyển giá trị đó thành số bằng hàm parseFloat.

+Rồi cộng tất cả tiền vào biến tongtienGH qua mỗi lần lặp.

* Sau cùng hiển thị giá trị tongtienGH thông qua thẻ có id = “tongtiengiohang”.

Text

Description automatically generated

* Hàm xoaCart() được thực hiện mỗi khi nhấn nút xóa cart.
* Trong hàm thực hiện xóa bỏ sessionStorage có tên cart.
* Hiển thị thẻ có id là giohang và tongtiengiohang là chuỗi rỗng.

Bước 3: Css cho đẹp

<style>

*#nut* {

        text-align: center;

    }

*#nut* button {

        width: 120px;

        height: 30px;

        background-color: darkcyan;

        color: whitesmoke;

    }

*#giohang* {

        width: 700px;

        margin: auto;

    }

*#tongtiengiohang* {

        text-align: center;

        font-size: 1.2em;

        color: darkgreen;

    }

*#giohang* *.rows* {

        display: grid;

        grid-template-columns: 50px 350px 100px 100px 90px;

        background-color: aquamarine;

    }

*#giohang* *.rows* \* {

        border: 1px solid navy;

        padding: 10px;

        text-align: center;

    }

*#giohang* *.rows* input {

        width: 60px;

        text-align: center;

    }

</style>

Kết quả:

Table

Description automatically generated

4. Trang thanh toán và kiểm lỗi thanhtoan.html

Bước 1:

<body>

    <div *class*="container">

        <h3>THANH TOÁN MUA HÀNG</h3>

        <form *action*="" *name*="frm1" *onsubmit*="return kiemtra()">

            <div *class*="name">

                <p>Họ tên</p>

                <input *type*="text" *placeholder*="Họ tên người nhận" *name*="hoten"><br>

                <span *id*="loi\_ht"></span>

            </div>

            <div *class*="email">

                <p>Email</p>

                <input *type*="email" *placeholder*="Email người nhận" *name*="email"><br>

                <span *id*="loi\_email"></span>

            </div>

            <div *class*="phone">

                <p>Số điện thoại</p>

                <input *type*="text" *placeholder*="Số điện thoại người nhận" *name*="sdt"><br>

                <span *id*="loi\_sdt"></span>

            </div>

            <div *class*="phuongthuc">

                <p>Phương thức thanh toán</p><br>

                <input *type*="radio" *name*="pay\_method"><label *for*="">Chuyển khoản</label>

                <input *type*="radio" *name*="pay\_method"><label *for*="">Thẻ ATM nội địa</label>

                <input *type*="radio" *name*="pay\_method"><label *for*="">Khi nhận hàng</label>

                <br>

                <span *id*="loi\_pay"></span>

            </div>

            <div *class*="tinhThanh">

                <p>Chọn tỉnh</p>

                <select *class*="chonTinh" *name*="tinh">

                    <option *value*="0">Chọn tỉnh</option>

                    <option *value*="An Giang">An Giang

                    <option *value*="Bà Rịa - Vũng Tàu">Bà Rịa - Vũng Tàu

                    <option *value*="Bắc Giang">Bắc Giang

                    <option *value*="Bắc Kạn">Bắc Kạn

                    <option *value*="Bạc Liêu">Bạc Liêu

                    <option *value*="Bắc Ninh">Bắc Ninh

                    <option *value*="Bến Tre">Bến Tre

                    <option *value*="Bình Định">Bình Định

                    <option *value*="Bình Dương">Bình Dương

                    <option *value*="Bình Phước">Bình Phước

                    <option *value*="Bình Thuận">Bình Thuận

                    <option *value*="Bình Thuận">Bình Thuận

                    <option *value*="Cà Mau">Cà Mau

                    <option *value*="Cao Bằng">Cao Bằng

                    <option *value*="Đắk Lắk">Đắk Lắk

                    <option *value*="Đắk Nông">Đắk Nông

                    <option *value*="Điện Biên">Điện Biên

                    <option *value*="Đồng Nai">Đồng Nai

                    <option *value*="Đồng Tháp">Đồng Tháp

                    <option *value*="Đồng Tháp">Đồng Tháp

                    <option *value*="Gia Lai">Gia Lai

                    <option *value*="Hà Giang">Hà Giang

                    <option *value*="Hà Nam">Hà Nam

                    <option *value*="Hà Tĩnh">Hà Tĩnh

                    <option *value*="Hải Dương">Hải Dương

                    <option *value*="Hậu Giang">Hậu Giang

                    <option *value*="Hòa Bình">Hòa Bình

                    <option *value*="Hưng Yên">Hưng Yên

                    <option *value*="Khánh Hòa">Khánh Hòa

                    <option *value*="Kiên Giang">Kiên Giang

                    <option *value*="Kon Tum">Kon Tum

                    <option *value*="Lai Châu">Lai Châu

                    <option *value*="Lâm Đồng">Lâm Đồng

                    <option *value*="Lạng Sơn">Lạng Sơn

                    <option *value*="Lào Cai">Lào Cai

                    <option *value*="Long An">Long An

                    <option *value*="Nam Định">Nam Định

                    <option *value*="Nghệ An">Nghệ An

                    <option *value*="Ninh Bình">Ninh Bình

                    <option *value*="Ninh Thuận">Ninh Thuận

                    <option *value*="Phú Thọ">Phú Thọ

                    <option *value*="Quảng Bình">Quảng Bình

                    <option *value*="Quảng Bình">Quảng Bình

                    <option *value*="Quảng Ngãi">Quảng Ngãi

                    <option *value*="Quảng Ninh">Quảng Ninh

                    <option *value*="Quảng Trị">Quảng Trị

                    <option *value*="Sóc Trăng">Sóc Trăng

                    <option *value*="Sơn La">Sơn La

                    <option *value*="Tây Ninh">Tây Ninh

                    <option *value*="Thái Bình">Thái Bình

                    <option *value*="Thái Nguyên">Thái Nguyên

                    <option *value*="Thanh Hóa">Thanh Hóa

                    <option *value*="Thừa Thiên Huế">Thừa Thiên Huế

                    <option *value*="Tiền Giang">Tiền Giang

                    <option *value*="Trà Vinh">Trà Vinh

                    <option *value*="Tuyên Quang">Tuyên Quang

                    <option *value*="Vĩnh Long">Vĩnh Long

                    <option *value*="Vĩnh Phúc">Vĩnh Phúc

                    <option *value*="Yên Bái">Yên Bái

                    <option *value*="Phú Yên">Phú Yên

                    <option *value*="Tp.Cần Thơ">Tp.Cần Thơ

                    <option *value*="Tp.Đà Nẵng">Tp.Đà Nẵng

                    <option *value*="Tp.Hải Phòng">Tp.Hải Phòng

                    <option *value*="Tp.Hà Nội">Tp.Hà Nội

                    <option *value*="TP  HCM">TP HCM

                </select>

                <br>

                <span *id*="loi\_tinh"></span>

            </div>

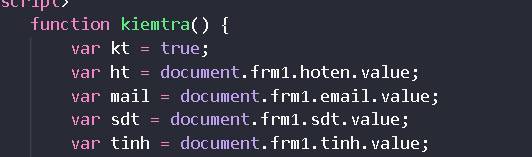
            <button *type*="submit" *onclick*="matchu()">Thanh toán</button>

        </form>

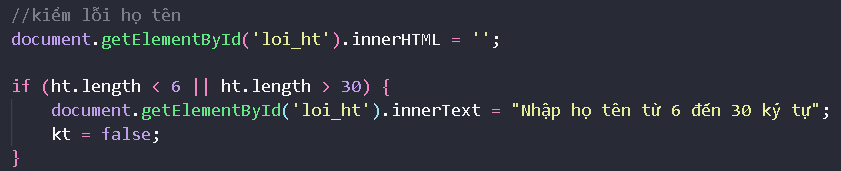
    </div>

</body>

* Trong body tạo thẻ div rồi tạo form, trong form thêm các thông tin muốn người dùng nhập vào.
* Đặt tên cho form là frm1 rồi cho sự kiện onsubmit trả về hàm kiemtra() để kiểm tra lỗi trong form.



* Trong hàm kiểm tra khai báo biến kt = true.
* Tạo các biến ht, mail, sdt, tinh để lấy giá trị trong form.



* Kiểm lỗi họ tên:

+ Cho thông báo lỗi họ tên là chuỗi rỗng.

+ Kiểm tra nếu chiều dài chuỗi nhập vào nhỏ hơn 6 hoặc lớn hơn 30 thì thông báo lỗi và trả về cho biến kt giá trị false.

Text

Description automatically generated

* Kiểm lỗi email:

+ Cho thông báo lỗi email là chuỗi rỗng trước.

+ Đặt biến mailformat theo các quy tắc chính quy kiểm tra email.

+ Điều kiện if test mail nhập vào có đúng định dạng theo mailformat không. Nếu không đúng thì hiện thông báo lỗi và trả biến kt về giá trị false.

Text

Description automatically generated

* Kiểm lỗi số điện thoại:

+ Đầu tiên cho lỗi số điện thoại bằng chuỗi rỗng.

+ Dòng if đầu tiên kiểm cho chuỗi nhập vào có 10 ký tự không, nếu độ dài khác 10 thì thông báo lỗi và trả biến kt về giá trị false.

+ Dòng if thứ 2 kiểm tra sdt có phải là số hay không, nếu không phải là số thì thông báo lỗi và trả về biến kt giá trị là false.

Text

Description automatically generated

* Kiểm tra lỗi thanh toán:

+ Cho thông báo lỗi thanh toán là chuỗi rỗng trước.

+ Cho biến pay trả về mảng bởi name ‘pay\_method’

+ Trong điều kiện if nếu 3 phần tử checked bằng false thì trả về lỗi phải chọn 1 phương thức thanh toán, và biến kt trả về giá trị false.

Text

Description automatically generated

* Kiểm tra lỗi chọn tỉnh:

+ Đầu tiên cho thông báo lỗi là chuỗi rỗng.

+ Nếu giá trị biến tỉnh trả về là 0 thì thông báo lỗi chọn tỉnh và biến kt trả về giá trị false.



Cuối hàm kiemtra() trả về giá trị biến kt là true hoặc false.

Text

Description automatically generated

Tạo 2 hàm bienmat() và matchu() để các dòng thông báo lỗi biến mất sau mỗi 5s khi click nút thanh toán.

A screenshot of a computer

Description automatically generated with medium confidence