**Họ và tên:** Huỳnh Nguyễn Quốc Bảo

**MSSV:** 22H1120002

**LAB JAVASCRIPT – MÔ TẢ**

**1. Kiểm tra form flogin**

- Trang **login.html** chứa form đăng nhập gồm 2 ô để nhập: Username & Password, và nút Login & trang **xulylogin.html** có nội dung tùy ý (không yêu cầu cấu hình cho phương thức post).

|  |  |
| --- | --- |
| <!DOCTYPE html> |  |
| <html lang="en"> |  |
| <head> |  |
| <meta charset="UTF-8"> |  |
| <meta name="viewport" content="width=device-width, initial-scale=1.0"> |  |
| <title>Login</title> | Tiêu đề trang hiển thị trên tab của trình duyệt. |
| <script type="text/javascript" src="checklogin.js"></script> | Chèn file **checklogin.js** vào file **login.html** để sử dụng các hàm javascript thuộc file tương ứng. |
| </head> |  |
| <body> |  |
| <form onsubmit="return checklogin();" id="**flogin**" name="**flogin**" method="**post**" action="xulylogin.html"> | Gọi hàm kiểm tra form: checklogin();  Sự kiện **onsubmit** xảy ra khi form được submit.  Hàm **checklogin()** được thực thi khi và chỉ khi nút submit được nhấn.  **ACTION:** chỉ định trang web sẽ nhận xử lý dữ liệu từ FORM này khi có sự kiện click của button SUBMIT, cụ thể là trang **xulylogin.html**.  **METHOD:** Xác định phương thức chuyển dữ liệu (POST)  Phương thức chuyển dữ liệu POST thực hiện gửi (lại) dữ liệu mà người dùng nhập vào form mỗi khi họ nhấn nút loại “submit” đến trang **xulylogin.html** và thực hiện các tác vụ bên trong file này. |
| Username: <br/> <input **name="u"** id="u" type="text" /> <br/> Password: <br/> <input name="p" id="p" type="password" /> <br/> | Tạo ra 2 ô Username và Password để người dùng có thể nhập tên đăng nhập và mật khẩu của họ. |
| <input type="submit" name="sub" id="sub" value="Login" /> | Nút login với kiểu submit – tức là sau khi nhấn nút thì dữ liệu được thu thập hoặc không, hoặc dẫn đến một sự kiện nào đó. |
| </form> |  |
| </body> |  |
| </html> |  |

- File **checklogin.js**, dùng để kiểm tra xem người dùng đã điền đầy đủ thông tin vào các ô Username & Password hay chưa, nếu một trong hai chưa thì thực hiện các tác vụ tương ứng.

|  |  |
| --- | --- |
| function checklogin() { |  |
| if (document.flogin.u.value == '') { | Lấy giá trị nơi thẻ input có name="**flogin**" của thẻ <form>, name="**u**" của thẻ <input> thuộc Username và value thể hiện việc lấy giá trị tại ô Username. (document.formName.elementName.propertyName)  Sau đó so sánh với chuỗi rỗng (Tức kiểm tra xem người dùng có phải đang để trắng ô Username và không nhập vào ô bất kỳ ký tự nào hay không) |
| alert("Bạn chưa nhập username"); | Nếu điều kiện đúng thì hiển thị hộp thoại thông báo. |
| document.flogin.u.focus();  //Chọn đối tượng u | **focus()** là phương thức được sử dụng để đặt focus (đưa con trỏ vào) phần tử, cho phép người dùng bắt đầu nhập vào phần tử đó, cụ thể là phần tử <input> của Username.  *Phương thức này chỉ hoạt động trên các phần tử nhập liệu (như input, textarea, select); không có nhiều phần tử khác nhau cùng được focus cùng một lúc.*  (*element*.focus()) |
| return false; |  |
| } |  |
| if (document.flogin.p.value == '') { | Tương tự với Username, kiểm tra xem người dùng có để trống ô Password không. |
| alert("Bạn chưa nhập password"); | Nếu điều kiện đúng thì trả về hộp thoại thông báo. |
| document.flogin.p.focus();  //Chọn đối tượng p | Đặt focus (đưa con trỏ vào) phần tử <input> của Password, cho phép người dùng bắt đầu nhập vào phần tử đó. |
| return false; |  |
| } |  |
| return true; //Cho submit form khi đã kiểm tra xong | Trả về true nếu cả 2 khối lệnh if trên đều không thỏa mãn. Giá trị true/false được trả về cho đối tượng form ở login.html. |
| } |  |

**2. Kiểm tra form search**

- Trang **tim.html** chứa form để nhập dữ liệu (từ khóa) và thực hiện tác vụ dựa trên nội dung mà người dùng vừa nhập & trang **kqtim.html** có nội dung tùy ý.

|  |  |
| --- | --- |
| <!DOCTYPE html> |  |
| <html lang="en"> |  |
| <head> |  |
| <meta charset="UTF-8"> |  |
| <meta name="viewport" content="width=device-width, initial-scale=1.0"> |  |
| <title>Tìm</title> | Tiêu đề trang hiển thị trên tab của trình duyệt. |
| <script type="text/javascript" src="checksearch.js"></script> | Chèn file **checksearch.js** vào file **tim.html** để sử dụng các hàm javascript thuộc file tương ứng. |
| </head> |  |
| <body> |  |
| <form **onsubmit**="return checksearch();" id="formtim" name="formtim" action="kqtim.html" method="**get**"> | Gọi hàm kiểm tra form: **checksearch();**  Sự kiện **onsubmit** xảy ra khi form được submit.  Hàm **checksearch()** được thực thi khi và chỉ khi nút submit được nhấn.  Khi sử dụng phương thức GET, *dữ liệu từ các trường input trong form (đối số) sẽ được gắn vào cuối URL* ***của thuộc tính Action trong tag <Form>*** dưới dạng *chuỗi truy vấn.*  kqtim.html?**tukhoa=aaaa**&**btntim=TÌM** (tukhoa & btntim ở 4 câu lệnh phía dưới)  🡪 dữ liệu form sẽ không được bảo mật  **GET** thường sử dụng để *lấy dữ liệu từ máy chủ* (tìm kiếm, truy xuất thông tin, điều hướng); *Không phù hợp cho các hành động có thể thay đổi dữ liệu trên máy chủ* (tạo mới/ cập nhật, thường sử dụng POST). |
| <input  **onclick**= "if(this.value == 'Tìm kiếm') **this**.value=''"  **onblur**= "if(this.value=='') **this**.value='Tìm kiếm'"  **name="tukhoa"** id="tukhoa" type="text" value="Tìm kiếm" /> | **Sự kiện onclick** được thực thi khi người dùng tác động click lên đối tượng trên form  **name="tukhoa":** Tên của trường sẽ được gửi đi khi form được submit.  **value="Tìm kiếm":** Giá trị ban đầu khi trang được tải lên của trường là "Tìm kiếm".  **if(this.value == 'Tìm kiếm') this.value=''**: Khi người dùng nhấp chuột vào trường input, JS sẽ kiểm tra giá trị của trường. *Nếu giá trị của trường là "Tìm kiếm" (giá trị mặc định), nó sẽ xóa chữ “Tìm kiếm” đang hiển thị trong ô & thay đổi giá trị thành chuỗi rỗng* – một vùng ô rỗng để người dùng nhập dữ liệu vào.  **onblur:** Sự kiện này được kích hoạt khi trường **input** mất focus, tức là người dùng nhấp chuột ra ngoài trường này (ra ngoài ô nhập liệu).  **if(this.value=='') this.value='Tìm kiếm'**: Khi người dùng ra khỏi trường input mà không nhập gì (giá trị của trường rỗng), JS sẽ đặt lại giá trị của trường là "Tìm kiếm", giúp giữ lại giá trị mặc định nếu người dùng không nhập nội dung. |
| <input **name="btntim"** id="btntim" type="submit" value="TÌM" /> | Khi người dùng nhập xong từ khóa và nhấn nút "TÌM", form sẽ được gửi đi với *giá trị từ khóa mà người dùng nhập* |
| </form> |  |
| </body> |  |
| </html> |  |

- Hàm kiểm tra form search trong file **checksearch.js**

|  |  |
| --- | --- |
| function checksearch() { |  |
| if (**document.formtim.tukhoa.value** == "Tìm kiếm") { | Nếu như bấm submit & *đúng là người dùng chưa nhập gì vào ô tìm kiếm* (nghĩa là giá trị của trường input có **name = “tukhoa”** là "**Tìm kiếm**") thì thực hiện các lệnh thuộc khối bên dưới. |
| alert("Bạn chưa nhập từ khóa tìm kiếm..."); | Nếu điều kiện đúng thì hiển thị hộp thoại thông báo. |
| document.formtim.tukhoa.**focus**(); | Đặt **focus** (đưa con trỏ vào) phần tử <**input**>, cho phép người dùng bắt đầu nhập vào dữ liệu để tìm kiếm. |
| return false; //Chặn form không cho submit |  |
| } | Tuy nhiên có thể gây ra điều vô lý, nếu thực tế người dùng nhập vào “Tìm kiếm” và nhấn tìm, thì vô tình điều kiện của if đúng thì nó sẽ không cho phép thực hiện tìm kiếm và thực hiện các câu lệnh trong if. |
| return true; //Cho submit form khi đã kiểm tra xong | Trả về giá trị cho sự kiện **onsubmit** của form. |
| } |  |

**3. Dò độ phân giải**

- Trang “index0.html” & các trang index1.html, index2.html, index3.html với các nội dung tùy ý & khác nhau (để dễ phân biệt).

- Thuộc tính **screen.width** chỉ trả về **độ phân giải màn hình** **của thiết bị,** *không phải kích cỡ của cửa sổ trình duyệt*. Vì thế nếu ta thử thay đổi kích cỡ cửa sổ trình duyệt thì ta sẽ không nhận được sự thay đổi nào ngoài nội dung của trang **index0.html.**

- Đoạn code này mô tả cách thức trang web được hiển thị trên các kích cỡ màn hình khác nhau.

|  |  |
| --- | --- |
| <!DOCTYPE html> |  |
| <html lang="en"> |  |
| <head> |  |
| <meta charset="UTF-8"> |  |
| <meta name="viewport" content="width=device-width, initial-scale=1.0"> |  |
| <title>Dò độ phân giải</title> |  |
| <script type='text/javascript'> |  |
| if (screen.width == 800) { | Nếu độ rộng của màn hình có chiều rộng là 800px |
| document.location = 'index1.html'; | Hiển thị nội dung của trang 'index1.html' |
| } else if (screen.width == 1024) | Nếu độ rộng của màn hình có chiều rộng là 1024px |
| document.location = 'index2.html'; | Hiển thị nội dung của trang 'index2.html' |
| else document.location = 'index3.html'; | Nếu ngoài 2 độ rộng này thì hiển thị nội dung của trang 'index3.html' |
| </script> |  |
| </head> |  |
| <body> |  |
| </body> |  |
| </html> |  |

**4. Hiện giờ hiện hành**

- Trong thẻ body bổ sung đoạn code dưới đây. Đoạn code này giúp *hiển thị ngày, giờ hiện hành lên trang web.*

|  |  |
| --- | --- |
| <script type="text/javascript"> |  |
| function tg() { |  |
| var currentTime = new Date(); | Tạo đối tượng **Date** để lấy ngày giờ hiện tại và lưu vào biến **currentTime**. |
| var month = currentTime.getMonth() + 1; | Phương thức **getMonth()** của đối tượng Date trả về **chỉ số** của tháng, bắt đầu từ 0 cho tháng 1 (tháng Giêng) đến 11 cho tháng 12 (tháng Mười Hai). Vì thế cần +1 vào giá trị trả về của getMonth() để có được số tháng đúng theo cách tính thông thường. |
| var day = currentTime.getDate(); | Lấy ngày trong tháng (1 – 31). |
| var year = currentTime.getFullYear(); | Lấy năm. |
| var hour = currentTime.getHours(); | Lấy giờ (0 – 23). |
| var minutes = currentTime.getMinutes(); | Lấy phút. |
| var seconds = currentTime.getSeconds(); | Lấy giây. |
| document.**getElementById("dongho").innerHTML** = day + "/" + month + "/" + year + " " + hour + ":" + minutes + ":" + seconds; | Truy xuất thẻ có **id = “dongho”,** cụ thể là thẻ <span>& thay đổi (chèn) nội dung (mới) của thẻ này, đó là ngày giờ vừa lấy được từ các câu lệnh trên. |
| } |  |
| setInterval("tg()", 1000); | Hàm **setInterval** nhận hai tham số: *Tên hàm* cần gọi liên tục & *khoảng thời gian* giữa các lần gọi hàm (tính bằng **ms**).  Câu lệnh này thiết lập *hàm được gọi liên tục* sau mỗi khoảng thời gian **1000ms (1s).** |
| </script> |  |
| <span **id="dongho"**></span> | Đặt thẻ <**span**> để chèn được dữ liệu cần hiển thị ở hàm tg(). |

**5. Đếm giảm dần để chuyển trang**

- Trang **“chuyen.html”** & trang **“index.html”** có nội dung tùy ý.

- Đoạn code thực hiện việc: sau một khoảng thời gian (5s) nó sẽ chuyển từ “chuyen.html” sang “index.html”. Trước đó tại giao diện của trang “chuyen.html” hiển thị thời gian đếm ngược chuyển trang.

|  |  |
| --- | --- |
| <!DOCTYPE html> |  |
| <html lang="en"> |  |
| <head> |  |
| <meta charset="UTF-8"> |  |
| <meta name="viewport" content="width=device-width, initial-scale=1.0"> |  |
| <title>Chuyển trang</title> |  |
| </head> |  |
| <body> |  |
| <script> |  |
| var sogiay = 5; | Khởi tạo biến **sogiay** với giá trị là 5. |
| setTimeout("document.location = **'index.html'**", sogiay \* 1000); | **setTimeout** được sử dụng để thực hiện một tác vụ sau một khoảng thời gian nhất định.  Cụ thể tác vụ là thay đổi địa chỉ của trang hiện tại **(chuyent.html)** sang **index.html** (với lệnh **document.location**) sau *sogiay \* 1000 – cụ thể là sau 5s.* |
| setInterval("document.getElementById(**'sogiay'**).innerHTML = sogiay--", 1000); | **setInterval** được sử dụng để thực hiện một hành động lặp đi lặp lại *(Chèn số giây hiện tại của biến sogiay vào thẻ có id = “sogiay”, sau đó lấy sogiay-1)* sau mỗi khoảng thời gian là *1000ms = 1s.* |
| </script> |  |
| **<center>** | Thẻ <**center**> dùng để canh giữa nội dung nằm trong thẻ này *(Không hỗ trợ HTML5).* (có thể dùng ***text-align: center;*** của CSS). |
| <p>Đã thực thi xong</p> |  |
| <p><a href='index.html'>Quay lại trang chủ</a></p> |  |
| <p>Sẽ quay lại trang chủ sau <span **id="sogiay"**></span>&nbsp;giây nữa</p> | Số giây giảm dần từ 5 về 1 sẽ *được chèn vào thẻ <****span****>.* |
| **</center>** |  |
| </body> |  |
| </html> |  |