

```
4 <head>
5   <!--meta-->
6   <title></title>
7   <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">
8   <link rel="shortcut icon" href="/favicon.ico" type="image/x-icon">
9   <link rel="icon" href="/favicon.ico" type="image/x-icon">
10  <!--CSS-->
11  <link type="text/css" rel="stylesheet" href="css/materialize.min.css" media="screen, projection">
12  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css">
13  <link rel="stylesheet" href="/css/animate.css">
14  <link rel="stylesheet" href="css/theme.css">
15 </head>
```

Jquery

Write less, do more

JQUERY LÀ GÌ ?



- ❖ jQuery là 1 bộ thư viện được viết dựa trên ngôn ngữ JavaScript giúp đơn giản hoá việc viết mã lệnh JavaScript.
- ❖ jQuery là thư viện mã nguồn mở (hoàn toàn miễn phí sử dụng) ra đời tháng 8 2006.
- ❖ Với dung lượng nhẹ và rất dễ sử dụng jQuery nhanh chóng được phổ biến rộng rãi và đang được bình chọn là một trong các thư viện Javascript tốt nhất hiện nay.

- Thao tác DOM để get hoặc set các phần tử HTML,CSS đơn giản hơn
- Xử lý các sự kiện (event)
- Tạo các hiệu ứng kết hợp animation (effects)
- Kỹ thuật AJAX tương tác với backend (Asynchronous JavaScript and XML)

Cách sử dụng jquery

```
<head>  
<script src="jquery-1.12.0.min.js"></script>  
</head>
```

• Sử dụng từ jQuery CDN (Google)

```
<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>  
</head>
```

• Sử dụng từ jQuery CDN (Yahoo)

```
<head>  
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.12.0.min.js"></script>  
</head>
```

Chương trình jquery đầu tiên

```
<script >
  $(document).ready(function(){
    $('#btn-change').click(function(){
      $('.demo h3').html('CyberSoft Academy');
    })
  })
</script>

</head>
<body>

  <div class="demo">
    <h3>Lập trình trực tuyến Myclass</h3>
  </div>
  <button class="btn btn-success" id="btn-change">Thay đổi nội dung</button>
```

`$("selector").action()`

❖ Selector:

- Có thể là id,class,tag,attribute...
- Ví dụ :

`<input type="text" class="btn-class" id="btnID" >`

Sử dụng với ID:

`$("input")`: Dom đến thẻ thông qua thuộc tính tagName

`$("# btnID")`: Dom đến thẻ thông qua thuộc tính id

`$(". btn-class")`: Dom đến thẻ thông qua className

`$("input[type='text']")`: Dom đến thẻ thông qua thuộc attribute

\$("selector").action()

❖ Selector:

Ngoài ra còn có thể sử dụng nhiều cách để Dom đến thẻ thông qua selector

Tham khảo thêm tại các trang sau:

<https://freetuts.net/selector-trong-jquery-574.html>

https://www.w3schools.com/jquery/jquery_ref_selectors.asp

\$(“selector”).action()

❖ **Action:**

- Là các hành động trên các thẻ.
- Ví dụ:
 - Action là các sự kiện (events) như click(), blur(), change() ... dùng để cài đặt xử lý khi người dung tương tác.
 - Action dùng để tạo hiệu ứng: fadeout(), fadeIn(), slideDown(),.....
 - Action dùng để get hoặc set nội dung html,css, attr, ...
 - Html() : get hoặc set thành phần innerHTML của thẻ
 - Val() : get hoặc set thành phần value của các thẻ input.
 - Attr() : get hoặc set các thuộc tính của thẻ.

Bài tập jquery



Đã tải lên 10/12 - YouTube

Gọi tên em trong đêm

Document

file:///D:/CYBERSOFT/cybersoft/DỰ%20ÁN%20THỰC%20TẾ/JAVASCRIPT/quan-ly-trung-tam/app/Views/index.html

Apps Học Javascript cơ bản Chuyển nhượng ER ERDPlus JS Hai cách lấy vị trí ngu Modernizr: the featur S Button | Semantic UI Material Design for B U Lorem Picsum Javascript nâng cao Javascript sida | Từ c

Quản lý Trung Tâm

Danh sách người dùng

Danh sách khóa học

Thống Kê

nhập tên người dùng...

Search

Thêm người dùng

Xóa người dùng

Modal

Tài Khoản	Mật Khẩu	Họ Tên	Email	Số điện thoại
<input type="checkbox"/> dangtrunghieu	01634643124	Đặng Trung Hiếu	dangturnghieu147@gmail.com	01634643124

Windows

Search

Taskbar icons

System tray

Bài tập jquery



Đã tải lên 10/12 - YouTube

Gọi tên em trong đêm

Document

file:///D:/CYBERSOFT/cybersoft/DỰ%20ÁN%20THỰC%20TẾ/JAVASCRIPT/quan-ly-trung-tam/app/Views/index.html

Apps Học Javascript cơ bản Chuyển nhượng ER ERDPlus JS Hai cách lấy vị trí ngu Modernizr: the featur S Button | Semantic UI Material Design for B U Lorem Picsum Javascript nâng cao Javascript sida | Từ c

Quản lý Trung Tâm

Danh sách người dùng

Danh sách khóa học

Thống Kê

nhập tên người dùng...

Search

Thêm người dùng

Xóa người dùng

Modal

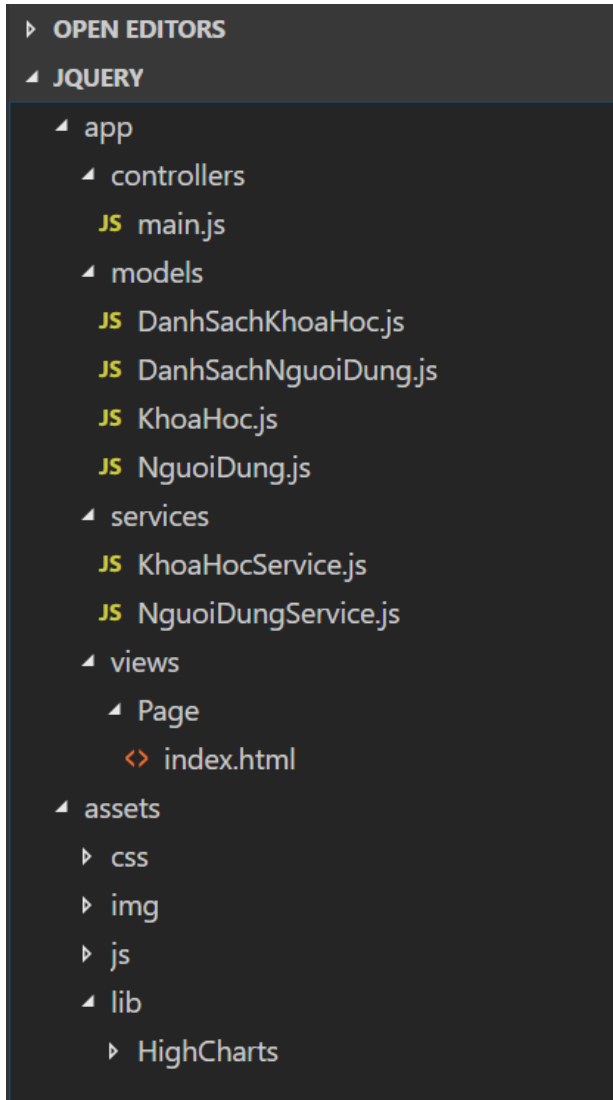
Tài Khoản	Mật Khẩu	Họ Tên	Email	Số điện thoại
<input type="checkbox"/> dangtrunghieu	01634643124	Đặng Trung Hiếu	dangturnghieu147@gmail.com	01634643124

Windows

Search

Taskbar icons

System tray



➤ Giới thiệu về kiến trúc dự án

Folder app: Chứa mọi xử ứng dụng web

- Controllers: Chứa các file js xử lý cho các page(view html) tương ứng
- Models: Chứa các class (prototype)
- Services: Chứa các kết nối liên kết với backend
- Views: Chứa các thiết kế giao diện html

Folder Assets: Chứa các tài nguyên của website như hình ảnh hay các thư viện js, css dùng chung cho toàn ứng dụng

- Css: Chứa các file css của các view page tương ứng
- Img: Chứa các hình ảnh của ứng dụng
- Js: Chứa các file js dùng chung cho toàn ứng dụng như validation V...V...
- Lib: Chứa các thư viện plugin từ bên ngoài sử dụng cho hệ thống

➤ Phần 1: Dàn layout áp dụng bootstrap 4

Bước 1: Import 1 số thư viện từ cdn bootstrap 4 cũng như đối với font-awesome hoặc tải về bỏ trong thư mục assets -> lib

Bước 2: Những phần liên quan đến bootstrap 4 cần áp dụng

- + Phần title chữ lớn: áp dụng class **display-4** trong bootstrap 4
- + Phần nav-tabs: Sử dụng **nav-tabs** bootstrap 4 (Quên có thể lên google gõ nav-tab link getbootstrap)
- + Thanh search: Dùng **input group** trong bootstrap 4 (Quên lên google search input group link v4-alpha hoặc getbootstrap)
- + Table: Dùng class **table** trong bootstrap 4 (Đặt id cho tbody để cần dùng hiển thị dữ liệu bằng jquery)
- + Open modal: Dùng **Modal** bootstrap (link w3school hoặc getbootstrap). Lưu ý: Để vào nhưng hiện tại chưa cần dùng

Kiến thức áp dụng cho bài tập



➤ Phần 2: Kiến thức javascript áp dụng

Bước 1: Xây dựng các **prototype** các class tương ứng theo yêu cầu phía dưới

Bước 2: Viết các phương thức **thêm, xóa, sửa, tìm kiếm** tương tự bài tập js hôm trước cho lớp đối tượng
(**Prototype** DanhSachNguoiDung)

Bước 3: Chèn các file script vào view index.html ở vị trí cuối cùng để đảm bảo rằng các tag html đã tạo xong

NguoiDung
▪ TaiKhoan : var
▪ MatKhau : var
▪ HoTen : var
▪ Email : var
▪ SoDT: var
▪ LoaiNguoiDung: var

DanhSachNguoiDung
▪ DSND: Array<NguoiDung>
• ThemSV()
• XoaSV()
• SuaSV()
• TimKiem()

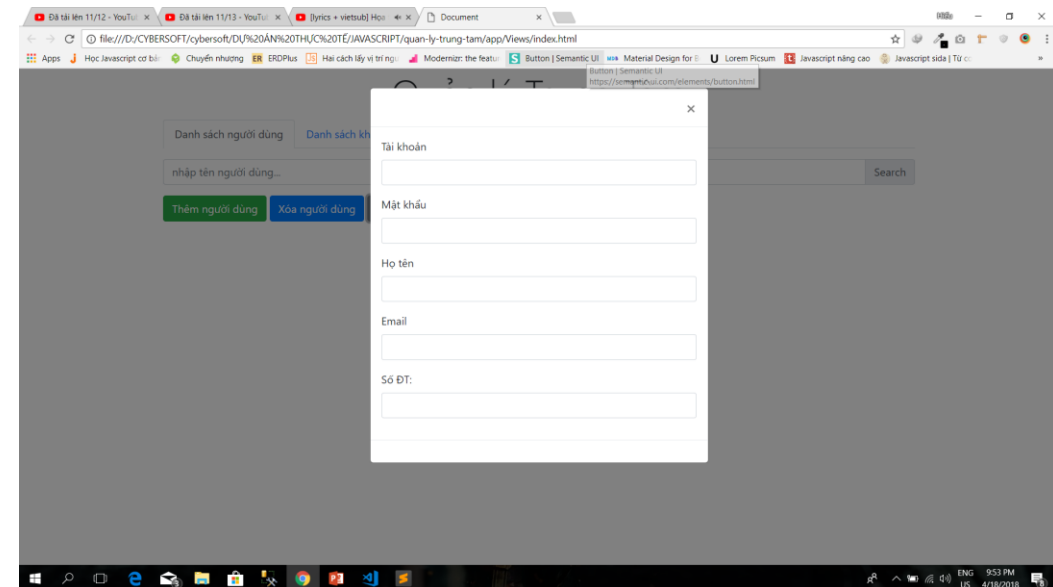
Kiến thức áp dụng cho bài tập

3.1 Chức năng thêm người dùng thực hiện viết bằng jquery

+ Bước 1: Tạo các thẻ input chứa thông tin người dùng nhập vào trên cửa sổ của popup modal (thiết kế input bằng bs4).

Lưu ý chỉ thiết kế sẵn phần **modal-body** (Phần **modal-header** và **modal-footer** ta dùng jquery lại sau).

```
<div class="modal fade" id="modal">
  <div class="modal-dialog">
    <div class="modal-content">
      <!--MODAL HEADER-->
      <div class="modal-header">
        <h3 class="modal-title"></h3>
        <button type="button" class="close" data-dismiss="modal">&times;</button>
      </div>
      <!--MODAL BODY-->
      <div class="modal-body">
        <div class="form-group">
          <label>Tên</label>
          <input type="text" class="form-control" id="txtTK">
        </div>
        <div class="form-group">
          <label>Mật khẩu</label>
          <input type="password" class="form-control" id="txtMK">
        </div>
        <div class="form-group">
          <label>Họ tên</label>
          <input type="text" class="form-control" id="txtHT">
        </div>
        <div class="form-group">
          <label>Email</label>
          <input type="text" class="form-control" id="txtEmail">
        </div>
        <div class="form-group">
          <label>Số ĐT:</label>
          <input type="text" class="form-control" id="txtSDT">
        </div>
      </div>
      <!--MODAL FOOTER-->
      <div class="modal-footer" id="modal-footer"></div>
    </div>
  </div>
</div>
```



3.1 Chức năng thêm người dùng thực hiện viết bằng jquery

+ Bước 2: Lấy thông tin người dùng nhập vào từ giao diện sử dụng các phương thức sau.

- Chức năng “Bấm nút thêm thì cửa sổ **popup modal** hiện ra”

* Thao tác xử lý:

+ Dùng jquery (**event**) Định nghĩa sự kiện **click()** cho nút **Thêm người dùng**

+ Trong sự kiện đó dùng (event) **trigger(“click”)** để gọi nút **modal click**

Ghi chú: Event trigger được dùng để gọi 1 sự kiện từ 1 thao tác sự kiện khác. Ở đây khi ta click nút **Thêm người dùng** thì nút **Modal** được click.

+ Dùng **Jquery dom** đến phần **.modal-title** đổi tên modal title thành Thêm thông tin người dùng.

Bằng cách sử dụng `$("#modal-title").html(“Thông tin người dùng”);`

+ Dùng **jquery dom** đến **.modal-footer** để thêm nút thêm thông tin người dùng.

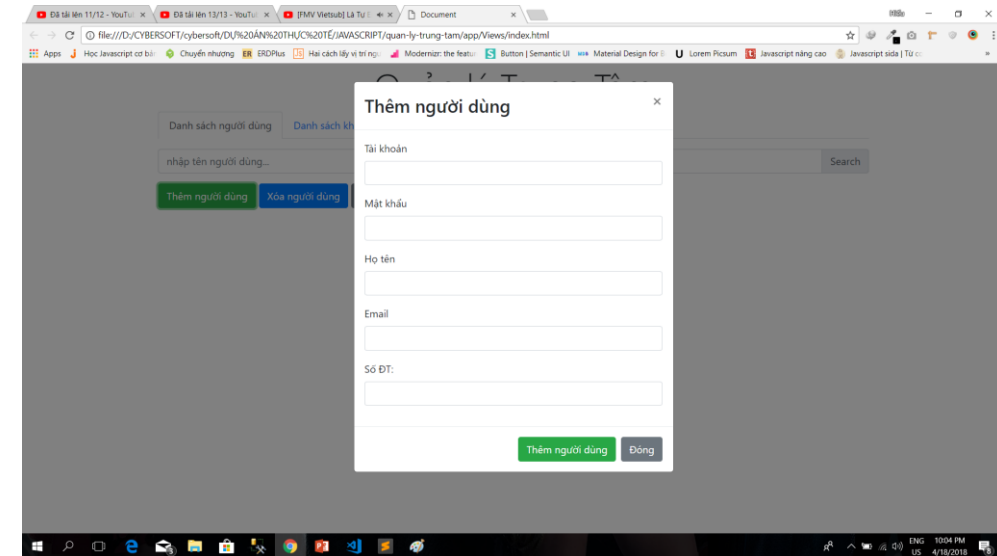
Kiến thức áp dụng cho bài tập

3.1 Chức năng thêm người dùng thực hiện viết bằng jquery

```
//Định nghĩa sự kiện click cho nút button Thêm Người Dùng bằng gọi ID
$('#btn-add-user').click(openModal);

//Xử lý cho sự kiện click Vào nút Thêm Người Dùng
function openModal(){
    //Tạo phần nội dung cho modalTitle
    var modalTitle = "Thêm người dùng";
    //Tạo phần nội dung cho modalFooter
    var modalFooter =
`<button class="btn btn-success" id="btnThem">Thêm người dùng</button>
<button class="btn btn-secondary" id="btnDong">Đóng</button>
`;

    //Set giá trị cho .modal-title và .modal-footer
    $('.modal-title').html(modalTitle);
    $('.modal-footer').html(modalFooter);
    //Gọi sự kiện trigger để click nút Modal
    $('#open').trigger('click')
}
```



3.1 Chức năng thêm người dùng thực hiện viết bằng jquery

- **Bước 3:** Xử lý nút **sự kiện** khi **nút thêm nhân viên** trong popup **click** thực hiện chức năng thêm nhân viên
- **Lưu ý:** `$("#btnThem").click()` sẽ không sử dụng được vì thành phần html của nút **input#btnThem** đó được add vào giao diện trong khi hàm main chúng ta đã truyền trước đó do đó jquery sẽ không thể tìm thấy nút ThemNhanVien đó. Do vậy trong trường hợp này ta sẽ sử dụng **action** là **delegate()**
- **Giải thích:** `$(“body”).delegate(“#btnThem”, “click”, function(){})` Cú pháp trên mang ý nghĩa xác định thẻ đã được tồn tại trước đó bao hàm các thẻ con sinh ra sau này trong đó. Thường dùng body vì body bao tất các các tag HTML từ lúc mới bắt đầu.

```
//Định nghĩa sự kiện click cho nút #btnThem được thêm vào modal-footer  
$('body').delegate('#btnThem', 'click', addUser);
```

3.1 Chức năng thêm người dùng thực hiện viết bằng jquery

- **Bước 3:** Xử lý nút **sự kiện** khi **nút thêm nhân viên** trong popup **click** thực hiện chức năng thêm nhân viên
- **Lưu ý:** `$("#btnThem").click()` sẽ không sử dụng được vì thành phần html của nút **input#btnThem** đó được add vào giao diện trong khi hàm main chúng ta đã truyền trước đó do đó jquery sẽ không thể tìm thấy nút ThemNhanVien đó. Do vậy trong trường hợp này ta sẽ sử dụng **action** là **delegate()**
- **Giải thích:** `$(“body”).delegate(“#btnThem”, “click”, function(){})` Cú pháp trên mang ý nghĩa xác định thẻ đã được tồn tại trước đó bao hàm các thẻ con sinh ra sau này trong đó. Thường dùng body vì body bao tất các các tag HTML từ lúc mới bắt đầu.

```
//Định nghĩa sự kiện click cho nút #btnThem được thêm vào modal-footer  
$('body').delegate('#btnThem', 'click', addUser);
```

3.1 Chức năng thêm người dùng thực hiện viết bằng jquery

- **Bước 4:** Dùng JQuery dom đến các **id** thẻ input lấy thông tin thông qua **action val()**. Ví dụ `$("#TaiKhoan").val()`; ngắn gọn hơn rất nhiều so với cú pháp `document.getElementById("TaiKhoan").value`. Sau đó đổ dữ liệu vào đối tượng gọi phương thức `themnguoidung` như javascript làm.

```
function addUser(){  
    //Lấy dữ liệu từ người dùng nhập vào thông qua action val()  
    var taikhoan = $('#txtTK').val();  
    var matkhou = $('#txtMK').val();  
    var hoten = $('#txtHT').val();  
    var email = $('#txtEmail').val();  
    var sodt = $('#txtSDT').val();  
  
    //Tạo mới đối tượng người dùng với các tham số đầu vào  
    var nguoiDung = new NguoiDung(taikhoan,matkhou,hoten,email,sodt);  
  
    //Thêm người dùng vào danh sách người dùng  
    danhSachNguoiDung.themNguoiDung(nguoiDung);  
  
    //Dùng trigger gọi nút close để đóng modal popup  
    $('.close').trigger('click');  
}
```

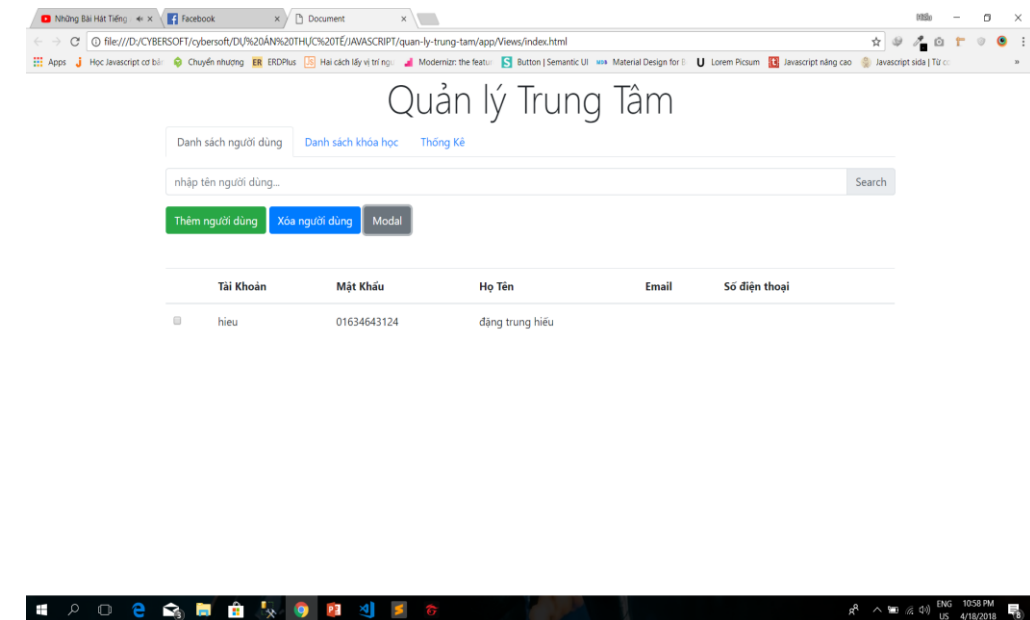
Kiến thức áp dụng cho bài tập

3.1 Chức năng thêm người dùng thực hiện viết bằng jquery

Bước 5: Viết phương thức load người dùng từ danh sách người dùng sử dụng jQuery.

Thay vì dùng đối tượng `document.create ...` ta dùng string template để làm trực tiếp các thẻ bằng chuỗi.

```
function loadNguoiDungTable(DSND)
{
    //Không dùng document thử dùng với jquery
    $("#tblDanhSachNguoiDung").html("");
    var noiDungTable = "";
    for(var i = 0 ; i < DSND.length ; i++)
    {
        var nguoiDung = DSND[i];
        noiDungTable += `
        <tr class="trNguoiDung">
            <td> <input type="checkbox" class="rdbTaiKhoan" value = "${nguoiDung.TaiKhoan}"> </td>
            <td> ${nguoiDung.TaiKhoan} </td>
            <td> ${nguoiDung.MatKhau} </td>
            <td> ${nguoiDung.HoTen} </td>
            <td> ${nguoiDung.Email} </td>
            <td> ${nguoiDung.SoDienThoai}</td>
        </tr>
        `;
    }
    $("#tblDanhSachNguoiDung").html(noiDungTable);
}
```



Kiến thức áp dụng cho bài tập

Lưu ý: `\${object}` stringtemplate chỉ hỗ trợ các trình duyệt sau:

Browser support

- Desktop

Chrome	Edge	Firefox	Internet Explorer	Opera	Safari
41	(Yes)	34	No support	28	9

- Mobile

Chrome Android	Firefox Mobile	IE Mobile	Opera	Safari
41	34	No support	28	9

1. Thao tác DOM get, set html css (query selector html, css – traversing - misc)

val(): để lấy và gán giá trị thẻ input

html(): dùng để lấy và gán giá trị html của thẻ có thể đóng

css() : xét css cho các phần tử html

trigger(): gọi 1 event từ 1 selector khác

delegate(): định nghĩa cho 1 sự kiện của 1 selector được sinh ra sau bởi javascript

Stringtemplate: dùng để xử lý chuỗi trong javascript dễ dàng hơn (tạo thẻ append ...)

2. Xử lý sự kiện (Event)

click(): event xảy ra khi người dùng click vào đối tượng selector

Tham khảo thêm: https://www.w3schools.com/jquery/jquery_ref_selectors.asp

1. Tìm kiếm người dùng

+Event : `keyup()`, `keydown()`, `keypress()` : sự kiện khi người dùng thao tác trên phím

+Method js:

- **`trim()`**: Sử dụng loại bỏ khoảng trống đầu cuối của chuỗi
- **`toLowerCase()`**: chuyển chuỗi dữ liệu thành chữ thường.
- **`toUpperCase()`**: ngược lại với `toLowerCase()`
- **`Search()`**: Tìm kiếm chuỗi kí tự có nằm trong đoạn văn bản hay không.

1. Tìm kiếm người dùng

Bước 1: tạo phương thức tìm người dùng trong prototype DanhSachNguoiDung

```
//Tạo phương thức tìm người dùng trong prototype DanhSachNguoiDung
this.timNguoiDung = function(value){
    //Tạo một danh sách mới chứa người dùng được tìm kiếm
    var danhSachTimKiem = new DanhSachNguoiDung();
    //chuyển từ khóa về chữ thường
    var tukhoa = value.toLowerCase();
    //Loại bỏ khoảng trắng 2 đầu từ khóa
    tukhoa = tukhoa.trim();
    //Tiến hành kiểm tra
    for( var i =0;i<this.DSND.length;i++){
        var nguoiDung = this.DSND[i];
        if(nguoiDung.HoTen.toLowerCase().trim().search(tukhoa) !== -1){
            danhSachTimKiem.themNguoiDung(nguoiDung);
        }
    }
    //Trả về danh sách tìm kiếm
    return danhSachTimKiem.DSND;
}
```

Bước 2:Viết function tìm người dung trong main.js

```
function searchNguoiDung(){
    //tạo mảng người dùng cần tìm
    var DSNDCanTim = [];
    //Tạo biến chữ từ khóa người dùng nhập vào
    var NDCanTim = $('#search').val();
    //kiểm tra nếu từ khóa rỗng thì xuất toàn bộ danh sách
    if(NDCanTim === ""){
        DSNDCanTim = danhSachNguoiDung.DSND;
    }
    // ngược lại xuất ra danh sách người dùng tìm kiếm
    else{
        DSNDCanTim = danhSachNguoiDung.timNguoiDung(NDCanTim);
        hightLight(NDCanTim);
    }
    TaoBang(DSNDCanTim);
}
```


2. Lưu trữ người dung vào localStorage

+Method js:

-**JSON.stringify(object)**: Chuyển 1 object javascript về dạng chuỗi

-**JSON.parse(chuoi)**: Chuyển 1 chuỗi về object javascript

+ Dùng localStorage:

+**setItem("Key",value)**: Lưu 1 chuỗi với từ khóa là **Key** và giá trị là **value** vào localStorage

+**getItem("Key")**: Lấy 1 chuỗi kết quả từ storage ra ngoài

+ **removeItem("Key")**

Các chức năng mở rộng của dự án



2. Lưu trữ người dùng vào localStorage

Bước 1: viết hàm lưu Storage

```
function lưuStorage(key,value){  
    //Chuyển json về chuỗi string  
    var stringValue = JSON.stringify(value);  
    //tiến hành lưu vào localStorage  
    localStorage.setItem(key,stringValue);  
}
```

Bước 3:Lấy dữ liệu từ localStorage

```
$('#btn-get-data').click(function(){  
    var DS = JSON.parse(localStorage.getItem('DSND'));  
    danhSachNguoiDung.DSND = DS;  
    TaoBang(danhSachNguoiDung.DSND);  
})
```

Bước 2:gọi hàm lưuStorage khi thêm người dùng

```
function addUser(){  
    var taikhoan = $('#txtTK').val();  
    var matkhau = $('#txtMK').val();  
    var hoten = $('#txtHT').val();  
    var email = $('#txtEmail').val();  
    var sodt = $('#txtSDT').val();  
    var nguoiDung = new NguoiDung(taikhoan,matkhau,hoten,email,sodt);  
    danhSachNguoiDung.themNguoiDung(nguoiDung);  
    lưuStorage('DSND',danhSachNguoiDung.DSND)  
    $('.close').trigger('click');  
    TaoBang(danhSachNguoiDung.DSND);  
    $('#modal-body input').val('');
```

Các chức năng mở rộng của dự án



3. Thực hiện tính năng xóa người dùng

+Jquery duyệt phần tử(Jquery traversing Method)

-\$('Selector').each(): phương thức duyệt tất cả các phần tử html có cùng selector.

-\$('Selector').is(':checked'): Trả về giá trị là **true** hoặc **false**. Nếu selector thỏa điều kiện giá trị trong ngoặc.

+Method js:

splice(index,number): Hàm xóa phần tử trong mảng của javascript. Index là vị trí phần tử cần xóa và number là số lượng phần tử được xóa tính từ vị trí index.

4. Thực hiện tính năng chỉnh sửa

+Jquery get,set attribute

-\$('Selector').attr("attrName"): phương thức lấy giá trị từ attribute của selector.

-\$('Selector').attr("attrName","value"): phương thức thêm thuộc tính hoặc gán giá trị thuộc tính attribute.

-\$('Selector').parent(): Dom đến thẻ cha chứa nó

-\$('Selector').closest('selector'): Dom đến thẻ bao nó (có thể ra nhiều cấp)

-\$('Selector').find('selector'): Dom đến thẻ con trong nó.

Hiệu ứng JQuery

▪ Hiệu ứng ẩn hiện `hide()`, `show()`

Sau đây là cú pháp đơn giản cho phương thức **show()** trong jQuery:

`[selector].show(speed, [callback]);`

`[selector].hide(speed, [callback]);`

Miêu tả chi tiết về các tham số:

- **speed** – Tốc độ ẩn hiện, giá trị: ‘slow’, ‘normal’, ‘fast’, 100,200ms....
- **callback** – tham số tùy ý này biểu diễn một hàm để được thực thi bất cứ khi nào hiệu ứng hoàn thành; thực thi một lần cho mỗi hiệu ứng.
- Tham số có thể có hoặc không

Hiệu ứng JQuery

■ Ví dụ hiệu ứng ẩn hiện hide(), show()

```
$(document).ready(function(){  
    $("#hide").click(function(){  
        $("p").hide(1000);  
    });  
    $("#show").click(function(){  
        $("p").show('fast');  
    });  
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<p>If you click on the "Hide" button, I will disappear.
```

```
</p>
```

```
<button id="hide">Hide</button>
```

```
<button id="show">Show</button>
```

```
</body>
```

Hiệu ứng JQuery

■ hiệu ứng fadeIn(), fadeOut(), slideDown(), slideUp()

Ví dụ fadeIn(), fadeOut()

```
$(document).ready(function(){
    $("#btn-1").click(function(){
        $('p').fadeIn();
    });
    $("#btn-2").click(function(){
        $('p').fadeOut(1000);
    });
});
</script>
</head>
<body>

<p>Demonstrate fadeIn() with different parameters.</p>

<button id="btn-1">Click to fade in </button>
<button id="btn-2">Click to fade out </button>

</body>
```

Ví dụ slideUp(), slideDown()

```
<script>
$(document).ready(function(){
    $(".btn-1").click(function(){
        $("p").slideUp();
    });
    $(".btn-2").click(function(){
        $("p").slideDown(function(){
            $("h1").hide();
        });
    });
});
</script>
</head>
<body>

<p>Demonstrate fadeIn() with different parameters.</p>
<h1>Demonstrate fadeIn() with different parameters.</h1>

<button class="btn-1">Click to slide Up</button>
<button class="btn-2">Click to slide Down</button>

</body>
```

Hiệu ứng JQuery

- hiệu ứng `toggle()` , `fadeToggle()`, `slideToggle()`

Toggle là sự kết hợp giữa khái niệm ẩn hiện tuy nhiên ta chỉ định nghĩa 1 lần

`[selector].toggle(speed, [callback]);`

`[selector].fadeToggle(speed, [callback]);`

`[selector].slideToggle(speed, [callback]);`

Hiệu ứng JQuery

- hiệu ứng `toggle()` , `fadeToggle()`, `slideToggle()`

Toggle là sự kết hợp giữa khái niệm ẩn hiện tuy nhiên ta chỉ định nghĩa 1 lần

`[selector].toggle(speed, [callback]);`

`[selector].fadeToggle(speed, [callback]);`

`[selector].slideToggle(speed, [callback]);`

Hiệu ứng JQuery

■ Ví dụ hiệu ứng toggle() , fadeToggle(), slideToggle()

```
<script>
$(document).ready(function(){
    $(".btn-1").click(function(){
        $("h1").fadeToggle();
    });
    $(".btn-2").click(function(){
        $("h1").slideToggle()
    });
    $(".btn-3").click(function(){
        $("h1").toggle()
    });
});
</script>
```

```
</head>
```

```
<body>
```

```
<h1>Demonstrate fadeIn() with different parameters.</h1>
```

```
<button class="btn-1">Click to fade toggle</button>
```

```
<button class="btn-2">Click to slide toggle</button>
```

```
<button class="btn-3">Click to toggle</button>
```

```
</body>
```

▪ Hiệu ứng `animation()`;

`Selector.animate(params, [duration, easing, callback]);`

- **params** – Một map các thuộc tính CSS mà hiệu ứng sẽ di chuyển theo nó.
- **duration** – Đây là tham số tùy ý biểu diễn hiệu ứng sẽ chạy bao lâu.
- **callback** – Đây là tham số tùy ý biểu diễn một hàm để gọi một khi hiệu ứng hoàn thành (có thể truyền vào số millisecond như các ví dụ trên).

Hiệu ứng JQuery

▪ Hiệu ứng `animation()`;

```
<script>
$(document).ready(function(){
    $("button").click(function(){

        $("div").animate({
            left: '250px',
            height: '+=150px',
            width: '+=150px'
        }, 2000, function(){

        });
    });
});
</script>
```

các thuộc tính
css thay đổi

duration

Callback function

Viết plugin jquery tạo hiệu ứng



Dự án quản lý trung tâm

Danh sách người dùng

[Danh sách khóa học](#)

[Thống kê](#)

Tìm kiếm

Thêm người dùng +

Xóa người dùng 🗑️

	Tài khoản	Mật khẩu	Họ tên	Email	Số ĐT
<input type="checkbox"/>	nguyen van b	123465789	nguyễn văn b	vanb@gmail.com	1234567893
<input type="checkbox"/>	tranvanc	123456789	trần văn c12345	email123@gmail.com	1234567892

Launch demo modal

Hướng dẫn thực hiện:



1. Đặt tên class cho cột td họ tên trong hàm loadtable (Ví dụ: `<td class="tdHoTen">`)
2. Viết hàm xử lý highlight
 1. Tạo function đặt tên **HightLight**
 2. Trong function lấy thông tin từ khóa người dùng nhập vào từ **input#tukhoa**
 3. Chọn duyệt class tdHoTen = `$(".tdHoTen").each` jquery
 4. Lấy giá trị html của jquery tìm xem có chứa cụm từ = với cụm từ người dùng nhập vào không bằng cách sử dụng hàm **indexOf()** => trả về vị trí `!=-1`
 5. Ta dùng hàm **substring()** để tách từ khóa trong chuỗi đó ra rồi change lại = cách thêm class span
 6. Nếu có chứa ta sẽ change text phần đó lại thành `từ khóa`
 7. Css cho class highlight.

Ví dụ:

Từ khóa: **văn**

Cột td chứa html là **nguyễn văn a** => **nguyễn** `` **văn** `` **a**

3. Viết plugin jquery cho từ đó nhấp nháy

Cú pháp viết plugin cho jquery dùng cú pháp

```
$.fn.TenPlugin = function (){  
}  
  
$('.class').TenPlugin();
```

Text.indexOf('tukhoa'): trả về vị trí của cụm từ khóa nằm trong văn bản (**Text**) -> nếu không có trả về -1.

substring(0,soluongtu): Trả về 1 chuỗi văn bản từ vị trí bắt đầu (0) đến số lượng vị trí tiếp theo kể từ vị trí bắt đầu.

Ví dụ: 'CyberSoft'.substring(0,4) => Cyber

substring(vitri): Trả về 1 chuỗi từ vị trí cần lấy đến hết chuỗi.

Ví dụ: 'CyberSoft'.substring(4) => Soft

- *Highchart js là 1 thư viện js giúp ta vẽ các biểu đồ thống kê.*
- *Cách sử dụng:*
 - B1: Vào website: <https://www.highcharts.com> (Hoặc google search highcharts)*
 - B2: Vào tab demo -> HighCharts Demo -> JSFiddle*
 - B3: Chép demo vào customize với dữ liệu của mình*