



BÀI TẬP

CHUYÊN ĐỀ

Xây dựng ứng dụng web với NODE.JS



BÀI 1: Tổng quan về NodeJS



Mục tiêu chính:

Làm quen với NodeJS

1.1. Làm quen với NodeJS

- Mục tiêu:
 - Làm quen với NodeJS
- Yêu cầu:
 - Cài đặt NodeJS và thử sử dụng với ứng dụng đầu tiên được hướng dẫn trong bài học.
 - Làm quen với các ứng dụng dùng để code.

1.2. Làm quen với NodeJS

- Yêu cầu:
 - Thử demo đầu tiên về NodeJS
- Hướng dẫn:
 - Code trang demo.js

```
http = require("http");

http.createServer(function(req, res){
    res.end("Chạy NodeJS rồi nhé!");
}).listen("8888");
```

Tạo server web bằng Http



Mục tiêu chính:

Tạo server web bằng Http của NodeJS

1.3. Tạo server web với Http, lắng nghe từ port 8080:

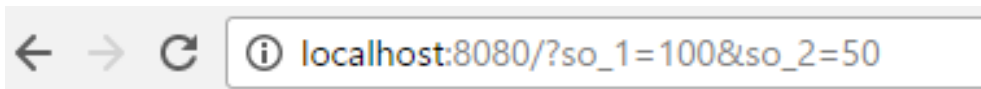
- Yêu cầu: thử nghiệm thực hiện tạo server web bằng http:
- Hướng dẫn:

```
//bài tập 2.1: tạo server http lắng nghe từ port 8080
http = require('http'),

http.createServer(function(req, res) {
    res.end("bạn đang vào website với port 8080");
}).listen(8080);
```

1.4. Tính tổng 2 số, với giá trị 2 số lấy từ đường dẫn:

- Yêu cầu: Cho người dùng nhập vào đường dẫn 2 biến như hình:



- Hướng dẫn code:

```
url = require('url'),
http = require('http'),

http.createServer(function(req, res) {
    var url_parts = url.parse(req.url,true);
    //console.log(url_parts.query);
    so_1 = url_parts.query.so_1;
    so_2 = url_parts.query.so_2;

    res.end("Kết quả của tổng 2 số: " + (so_1 * 1 + so_2 * 1));
}).listen(8080);
```

1.5. In bảng cửu chương từ biến post:

- Yêu cầu: Có thể dùng HTML form gửi đến server web của NodeJS, hoặc dùng postman để gửi dữ liệu form giả sang NodeJS.

- Hướng dẫn:

```
http = require('http'),
qs = require('querystring');
http.createServer(function(req, res) {
    var body='';
    req.on('data', function (data) {
        body +=data;
    });
    req.on('end',function(){
        var data_post = qs.parse(body);
        console.log(data_post);

        chuoai = '';
        if(typeof data_post != null){
            for(i = 1; i < 10; i++){
                bcc = data_post.bang_cuu_chuong;
                chuoai += bcc + ' x ' + i + ' = ' + (bcc * i) + "<br/>";
            }
        }
        else{
            chuoai = 'không có biến post';
        }
        res.end(chuoai);
    });
}).listen(8080);
```

BÀI 2: Module và Events



Mục tiêu chính:

*Áp dụng được cách sử dụng, xây dựng module trong NodeJS.
 Nắm bắt các vấn đề về Event*

2.1. Xây dựng module xử lý định dạng chữ

- Yêu cầu: Thực hiện một module định dạng chữ có 3 hàm xử lý như sau:
 - Hàm xử lý camel_case: Tất cả chuỗi đều chuyển về thành dạng như ví dụ sau: "xin chào các bạn" sẽ thành "XinChaoCacBan"
 - Hàm xử lý bỏ dấu tiếng việt: với tất cả các câu thành dạng như ví dụ: "Xin chào các bạn" sẽ chuyển thành "Xin chao cac ban"
 - Hàm xử lý tạo friendly URL từ các chuỗi tên: hàm này sẽ sử dụng hàm xử lý bỏ dấu tiếng việt trước rồi mới chuyển thành **Friendly** URL với ví dụ như sau: "Xin chào các bạn" sẽ chuyển thành "xin-chao-cac-ban"
- Hướng dẫn:
 - Hàm xử lý camel_case:

```
exports.camel_case = function(str){
    return str.replace(/(?:^|\w|[A-Z]|\b\w)/g, function(letter, index) {
        return index == 0 ? letter.toLowerCase() : letter.toUpperCase();
    }).replace(/\s+/g, '');
}
```

- Hàm xử lý bỏ dấu tiếng việt:

```
exports.bo_dau_tiem_viet = function(str){
  var in_chrs = 'àáâãäåçèéêëìíîïñòóôõöùúûüýÿÀÁÂÃÄÅÇÈÉÊËÌÍÎÏÑÒÓÔÕÖÙÚÛÜÝ',
    out_chrs = 'aaaaaaceeeeeeiíiíinooooouuuuyyAAAAACEEEEIIIINOOOOOUUUUY',
    chars_rgx = new RegExp('[' + in_chrs + ']', 'g'),
    transl = {}, i,
    lookup = function (m) { return transl[m] || m; };

  for (i=0; i<in_chrs.length; i++) {
    transl[ in_chrs[i] ] = out_chrs[i];
  }

  return str.replace(chars_rgx, lookup);
}
```

– Hàm xử lý friendly URL:

```
exports.friendly_url = function(str){
  str = this.bo_dau_tiem_viet(str);
  return str.toLowerCase()
    .replace(/^[^a-z0-9]+/g, "-")
    .replace(/^-+|-+$/g, "-")
    .replace(/^-+|-+$/g, '');
}
```

2.2. Xây dựng module dạng class đối tượng trong NodeJS:

- Yêu cầu: Thực hiện một class ngay_thang: có phương thức chuyển định dạng ngày với các yêu cầu chi tiết sau:
 - Phương thức khởi tạo: nếu người dùng có truyền ngày (kiểu dữ liệu Date trong javascript) vào thì gán thuộc tính ngay bằng biến truyền vào, còn nếu người dùng không truyền biến ngày vào thì sẽ lấy ngày hiện tại
 - Phương thức định dạng ngày tháng:
 - Có 2 biến truyền vào là biến ngôn ngữ, biến định dạng (rút gọn hay đầy đủ chữ).
 - Xử lý định dạng ngày như ví dụ sau:

nếu là tiếng Việt và rút gọn sẽ như sau: “dd/mm/yyyy”, nếu là tiếng việt và đầy đủ thì “Thứ 4 ngày 5 tháng 4 năm 2017”

nếu là tiếng Anh và rút gọn sẽ như sau: “mm/dd/yyyy”, nếu là tiếng anh và đầy đủ thì “Wednesday 5 April 2017”

- Hướng dẫn:
 - Xử lý định dạng ngày tháng theo yêu cầu:

```
switch(str) {
  case "vi":
    return (loai == "rut_gon")?
      (this.ngay.getDate()
      +'/'+(this.ngay.getMonth()+1)
      +'/' +this.ngay.getFullYear())
      :
      (days_vi[this.ngay.getDay()]
      + ' ngày ' + this.ngay.getDate()
      + ' ' + months_vi[this.ngay.getMonth()]
      + ' '+this.ngay.getFullYear());

    break;
  case "en":
    return (loai == "rut_gon")?
      ((this.ngay.getMonth()+1)
      +'/' +this.ngay.getDate()
      +'/' +this.ngay.getFullYear())
      :
      (days[this.ngay.getDay()]
      + ' ' + this.ngay.getDate()
      + ' '+months[this.ngay.getMonth()]
      + ' '+this.ngay.getFullYear());

    break;
}
```

2.3. Tạo sự kiện với EventEmitter:

- Yêu cầu: Tạo ra một Event lắng nghe bằng EventEmitter và gọi nó chạy khi trang được request trong Http server.
- Hướng dẫn:

```
http = require("http");

EventEmitter = require("events").EventEmitter;
my_event = new EventEmitter();

my_event.on("lang_nghe", function () {
    console.log("sự kiện đã được gọi");
});

http.createServer(function(request, response) {
    my_event.emit("lang_nghe");
}).listen(8080);
```

2.4. Tạo sự kiện với EventEmitter, và response trong Event mới tạo chứ không response trực tiếp từ Http server:

- Yêu cầu: Hoàn thiện một event lắng nghe dùng EventEmitter và cho phép nhận biến response từ Http server để trả lời.
- Hướng dẫn:

```
http = require("http");

EventEmitter = require("events").EventEmitter;
my_event = new EventEmitter();

my_event.on("lang_nghe", function (response) {
    console.log("sự kiện đã được gọi");
    response.end("chạy lệnh trong sự kiện lắng nghe");
});

http.createServer(function(request, response) {
    gia_tri = my_event.emit("lang_nghe", response);
}).listen(8080);
```


BÀI 3: Buffer và Stream



Mục tiêu chính:

Áp dụng stream và buffer tạo nên một ứng dụng đơn giản.

Có đủ kiến thức, nền tảng để tiếp tục hoàn thành các chức năng lớn hơn khi vào xây dựng một website với express framework ở bài sau.

3.1. Tạo ứng dụng tính tiền khi bán ngoại tệ và lưu lại danh sách giao dịch:

- Yêu cầu: Làm form giao diện đơn giản như sau:

Khi chưa nhập số tiền bán: (sau khi chọn Loại tiền => hiển thị luôn tỷ giá)

Cửa hàng ngoại tệ
Bán ra
Mua vào
Cập nhật tỷ giá
Thống kê giao dịch

Phiếu ghi nhận bán

Chọn loại tiền
USD

Đơn giá bán:
25000 VNĐ = 1 USD

Nhập số tiền cần bán

Số tiền thu vào

Ghi nhận

Khi nhập xong số tiền bán và click vào nút ghi nhận:

Cửa hàng ngoại tệ
Bán ra
Mua vào
Cập nhật tỷ giá
Thống kê giao dịch

Phiếu ghi nhận bán

Chọn loại tiền
USD

Đơn giá bán:
25000 VNĐ = 1 USD

Nhập số tiền cần bán
500

Số tiền thu vào
12500000

Ghi nhận

- Hướng dẫn:

STT	Yêu cầu	Hướng dẫn
1	Tạo giao diện	Thiết kế giao diện như hình
2	Xây dựng một module xử lý dữ liệu với file dữ liệu là json	Dùng các hàm đọc dữ liệu từ tập tin của thư viện fs trong NodeJS để lấy dữ liệu cũng như thêm dữ liệu Các lịch sử giao dịch sẽ được lưu vào file du_lieu/ds_giao_dich.json
3	Xây dựng ajax với các thao tác từ html sang NodeJS	Có thể dùng HTTP trong Javascript hoặc Ajax trong jQuery

3.2. Xây dựng trang xem lại lịch sử giao dịch đã bán:

- Yêu cầu: Giao diện xem lịch sử sẽ giống như sau:

Cửa hàng ngoại tệ	Bán ra	Mua vào	Cập nhật tỷ giá	Thống kê giao dịch
-------------------	--------	---------	-----------------	--------------------

Danh sách giao dịch

STT	Loại tiền	Số tiền bán	Đơn giá	Số tiền thu vào	Thời gian giao dịch
1	USD	10000	23770	237700000	20/4/2017 10:20:13
2	USD	500	22770	11385000	20/4/2017 10:25:27
3	USD	1200	22770	27324000	20/4/2017 11:1:34
4	USD	1200	25000	30000000	20/4/2017 11:2:31
5	USD	2000	25000	50000000	22/5/2017 15:46:24

- Hướng dẫn:

STT	Yêu cầu	Hướng dẫn
1	Tạo giao diện	Thiết kế giao diện như hình

2	Xây dựng thêm hàm lấy danh sách giao dịch trong file libraries/xl_du_lieu.js	Dùng thư viện fs trong NodeJS để lấy dữ liệu từ file du_lieu/ds_giao_dich.json
3	Xây dựng hàm tự động load danh sách giao dịch khi vào trang ở html	Viết hàm trực tiếp ở file app.js thông qua phân chia các url và hiển thị lên website

3.3. Xây dựng trang Cập nhật tỷ giá bán:

- Yêu cầu: Giao diện cập nhật tỷ giá sẽ giống như sau:

Cửa hàng ngoại tệ
Bán ra
Mua vào
Cập nhật tỷ giá
Thống kê giao dịch

Phiếu cập nhật tỷ giá

Chọn loại tiền

USD

Số tiền bán ra

25000

Số tiền mua vào

24900

Ghi nhận

- Hướng dẫn:

STT	Yêu cầu	Hướng dẫn
1	Tạo giao diện	Thiết kế giao diện như hình
2	Xây dựng thêm hàm cập nhật tỷ giá bán trong file libraries/xl_du_lieu.js	Dùng thư viện fs trong NodeJS cập nhật lại nội dung của file ty_gia.json
3	Xây dựng ajax với các thao tác từ html sang NodeJS	Có thể dùng HTTP trong Javascript hoặc Ajax trong jQuery

BÀI 4: Express Framework

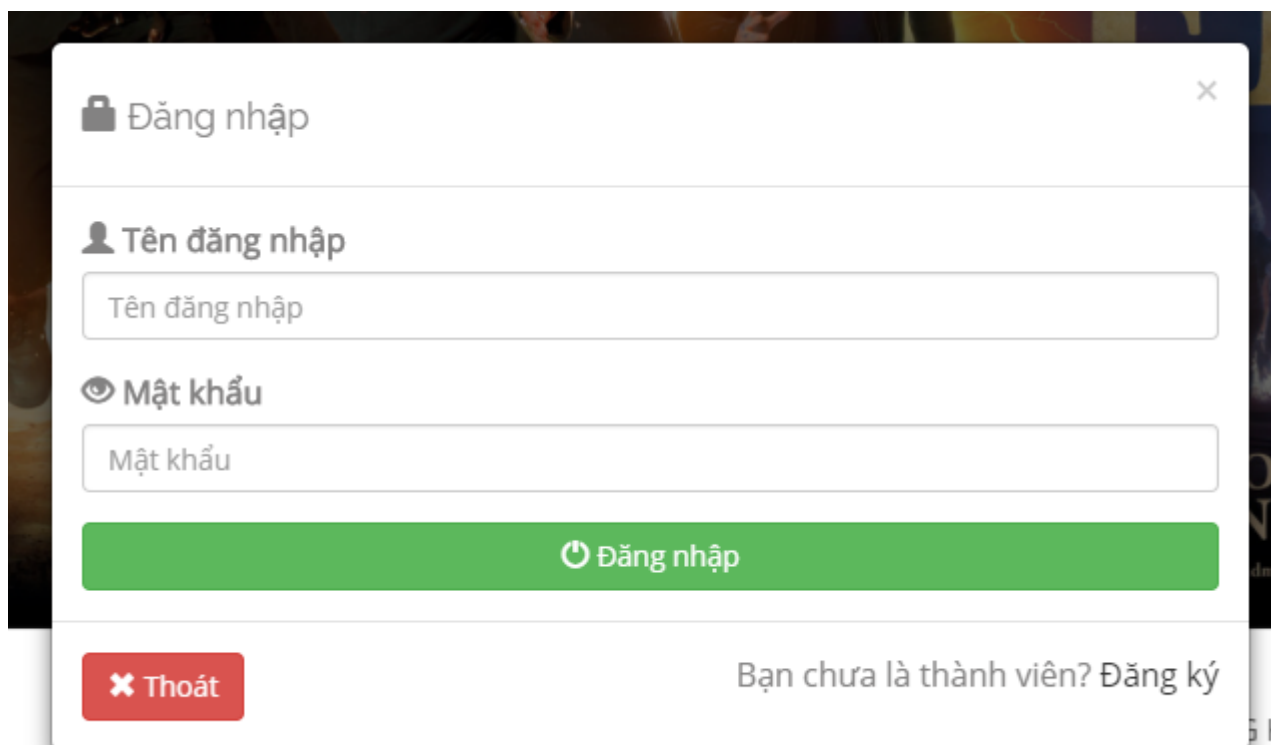


Mục tiêu chính:

Áp dụng Express Framework để thực hiện website một cách dễ dàng, nhanh chóng, hiệu quả trong NodeJS

4.1. Tạo Form và thực hiện đăng nhập người dùng:

- Yêu cầu:
 - + Tạo nút đăng nhập trên thanh menu
 - + Khi người dùng chọn vào nút đăng nhập thì Popup form đăng nhập hiện ra có giao diện:



- Hướng dẫn:

STT	Yêu cầu	Hướng dẫn
1	Tạo giao diện	Thiết kế giao diện như hình được cung cấp
2	Tạo Route cho gửi form đăng nhập bằng	Áp dụng bài Route trong Express tạo ra Route Post để có thể nhận dữ liệu từ form gửi qua



	Express	
3	Tạo file <code>nguoai_dung.js</code> trong thư mục	File <code>nguoai_dung.js</code> có nhiệm vụ như một controller chứa các hàm xử lý liên quan đến người dùng khi route chuyển qua.
4	Xử lý đăng nhập	Kiểm tra xem người dùng nhập dữ liệu vào có đúng với một chuỗi mặc định

4.2. Tạo Middleware đơn giản kiểm tra đăng nhập:

- Yêu cầu: Tạo một middleware chịu trách nhiệm kiểm tra xem người dùng đã đăng nhập hay chưa.
- Hướng dẫn:

STT	Yêu cầu	Hướng dẫn
1	Tạo một middleware trực tiếp trong file <code>app.js</code>	Tạo một middleware giả lập là đã đăng nhập rồi và cho hiển thị chuỗi tên người dùng mặc định

4.3. Tạo Middleware load danh sách menu lên trang web:

- Yêu cầu: Tạo một middleware đưa một mảng tạo sẵn trong code xử lý lên hiển thị trang cho tất cả các trang.

Và menu sẽ có giao diện như hình:

TRANG CHỦ VIDEO TIN TỨC ĐĂNG KÝ ĐĂNG NHẬP

- Hướng dẫn:

STT	Yêu cầu	Hướng dẫn
1	Tạo giao diện	Tạo giao diện bằng html thuần để hiển thị như hình
2	Xây dựng Middleware	Có thể tạo trực tiếp trong file <code>app.js</code> hoặc tạo một file riêng. Có nhiệm vụ load một mảng danh sách menu gán vào một biến để hàm xử lý khi route

		chuyển qua có thể dùng biến và hiển thị
--	--	---

4.4. Tạo Middleware hiển thị lỗi khi không tìm thấy trang:

- Yêu cầu: giao diện trang bị lỗi sẽ có như sau:



- Hướng dẫn:

STT	Yêu cầu	Hướng dẫn
1	Tạo giao diện	Dùng html, css để thực hiện trang hiển thị lỗi
2	Tạo middleware	Cót hể tạo middleware kiểm tra trực tiếp ở file app.js hoặc tạo một file middleware riêng.
3	Sử dụng middleware	Sử dụng middleware vừa tạo, xây dựng ở trên cho toàn bộ các trang

BÀI 5: EJS template

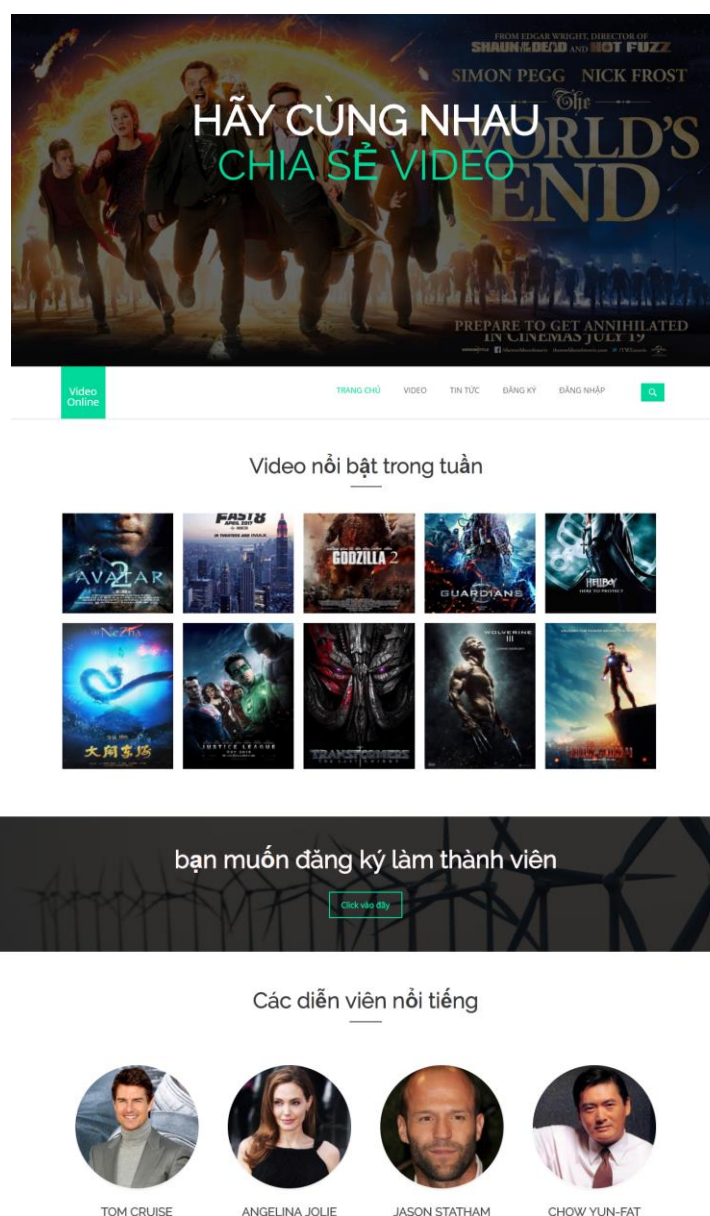


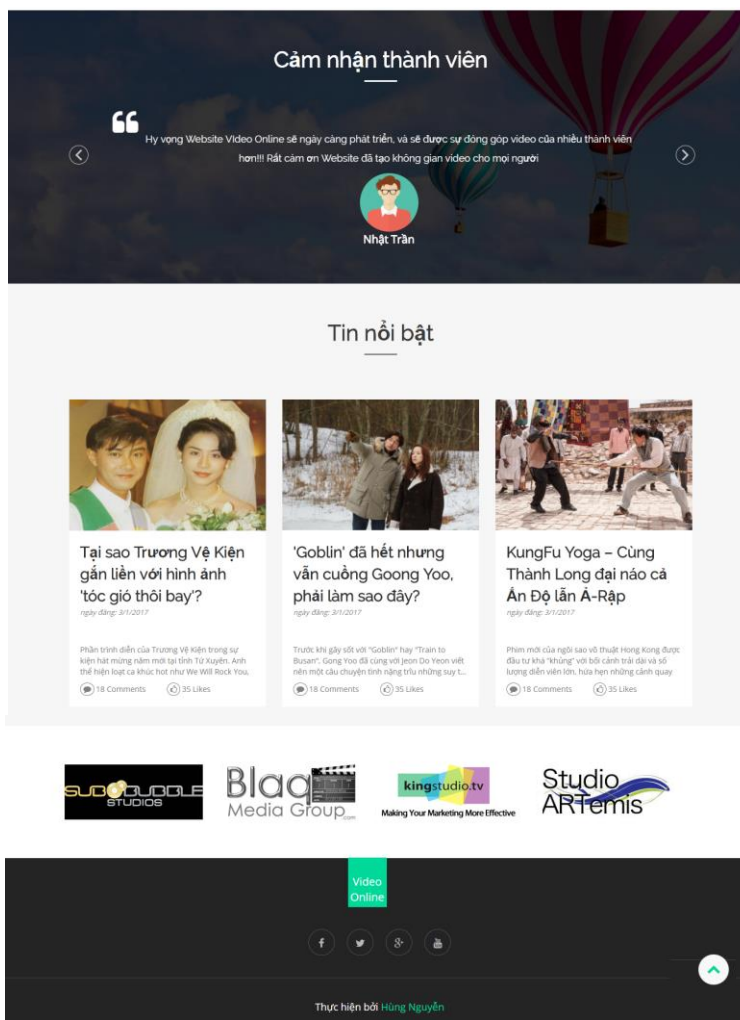
Mục tiêu chính:

Áp dụng được khả năng kế thừa trong template EJS, giúp cho xây dựng template website trở nên chuyên nghiệp hơn và khả năng bảo trì website sẽ dễ dàng khi cần thiết.

5.1. Tạo trang chủ như template:

- Yêu cầu:





- Hướng dẫn:

STT	Yêu cầu	Hướng dẫn
1	Tạo giao diện	Tạo giao diện trang chủ giống như hình được nêu trên.
2	Cắt template	Tạo các partials, module chia nhỏ từ trang để làm template => tăng tính tái sử dụng khi xây dựng template
3	Hiển thị trang chủ khi route gọi	Trong hàm xử lý khi route gọi qua, dùng lệnh res.render để load giao diện trang chủ hiển thị.

5.2. Tạo trang danh sách video:

- Yêu cầu: Có giao diện như sau:



Video
Online

TRANG CHỦ

VIDEO

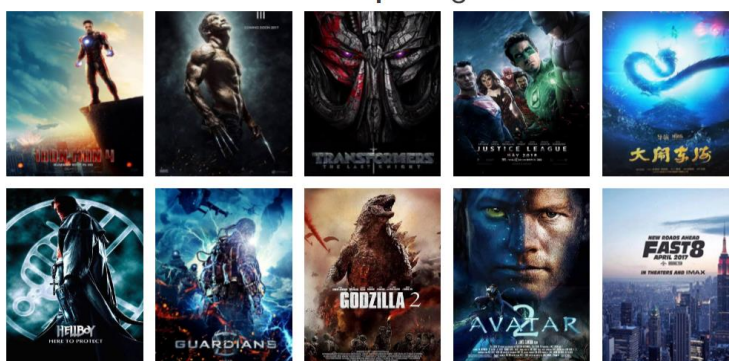
TIN TỨC

ĐĂNG KÝ

ĐĂNG NHẬP



Video nổi bật trong tuần



Video
Online



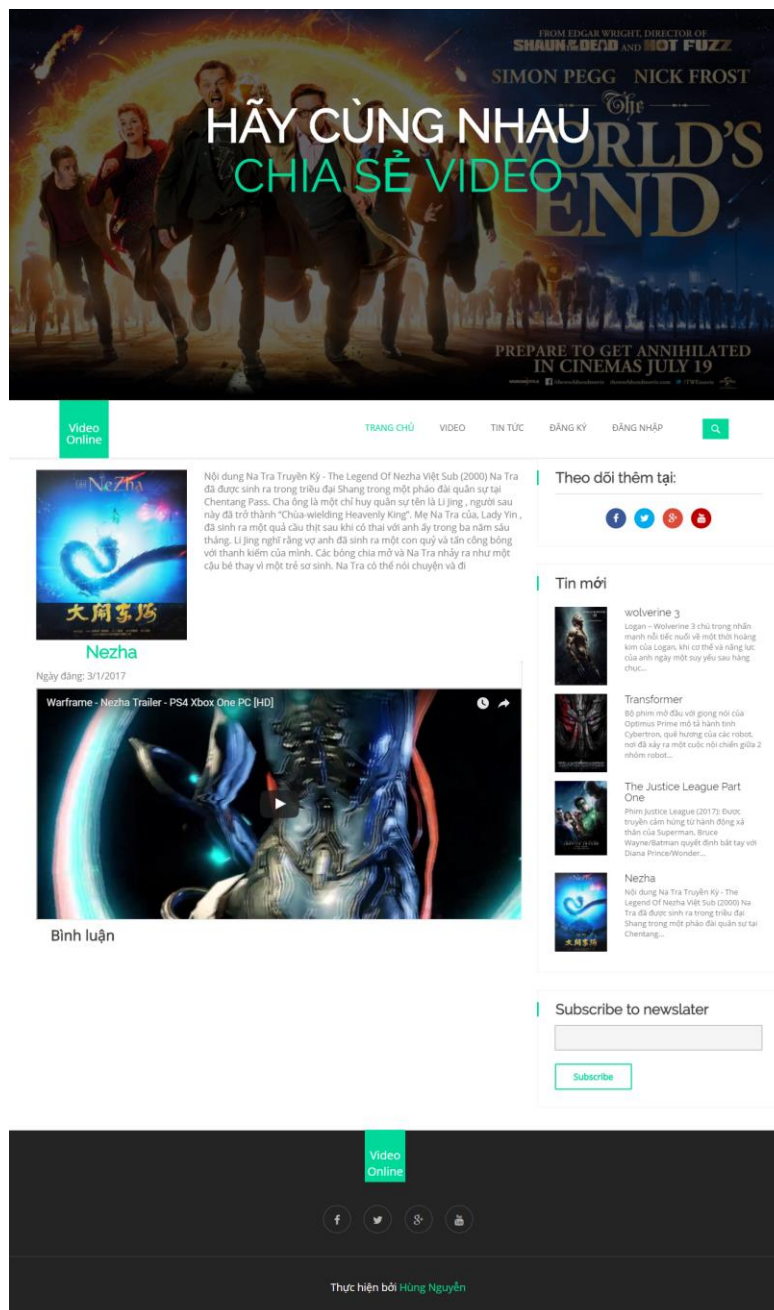
Thực hiện bởi Hùng Nguyễn

Hướng dẫn:

STT	Yêu cầu	Hướng dẫn
1	Tạo giao diện	Dùng html, css viết giao diện cho template trang danh sách video.
2	Route đường dẫn	Khi người dùng truy cập bằng đường dẫn domain/video sẽ cho phép vào trang danh sách video
3	Xây dựng hàm <code>ds_video</code> Trong file routes/video.js	Hàm có trách nhiệm xử lý load danh sách video (mảng giả) lên và gọi hàm chọn template chạy giao diện đã tạo.

5.3. Xây dựng trang chi tiết video:

- Yêu cầu: Thiết kế giao diện như hình:



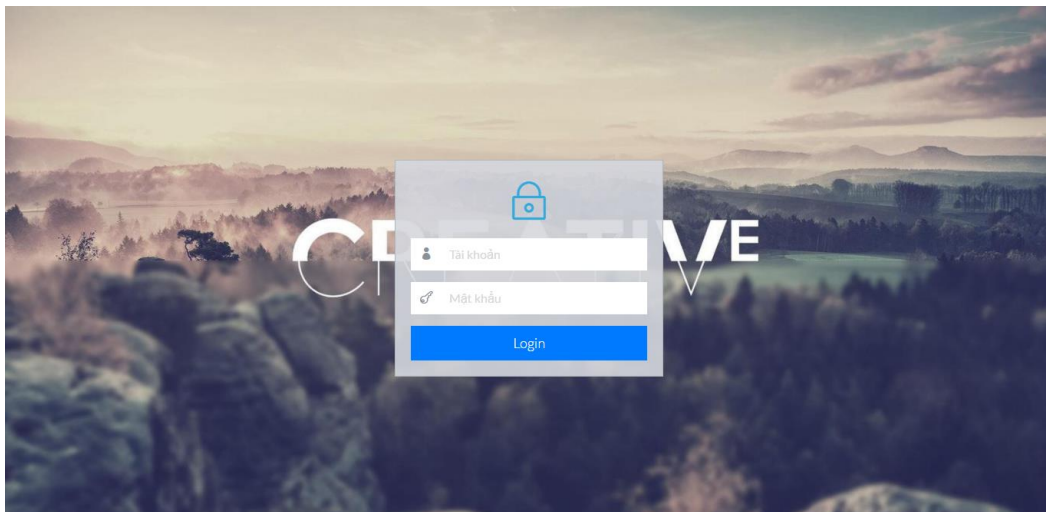
- Hướng dẫn:

STT	Yêu cầu	Hướng dẫn
1	Thực hiện giao diện	Sử dụng HTML, CSS để tạo giao diện giống như hình đã cho
2	Tạo route trong	Route đường dẫn domain/video/(id_video) để

	Express	có thể vào được trang giao diện
3	<p>Tạo hàm xử lý chi_tiet_video</p> <p>Trong file routes/video.js</p>	<p>Lấy dữ liệu giá thông tin một video và hiển thị lên giao diện.</p> <p>Ngoài ra còn tạo một mảng giả danh sách video mới và cũng load lên trên giao diện.</p>

5.4. Giao diện login admin:

- Yêu cầu: Thiết kế giao diện như hình:

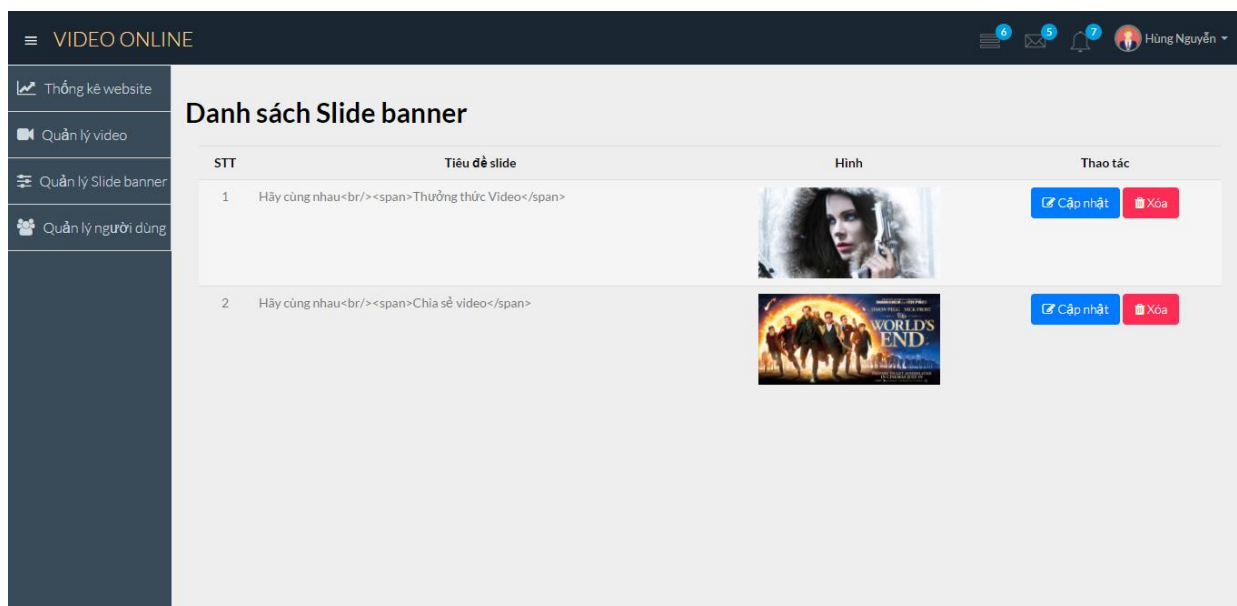


- Hướng dẫn:

STT	Yêu cầu	Hướng dẫn
1	Thực hiện giao diện	Sử dụng HTML, CSS để tạo giao diện giống như hình đã cho
2	Tạo route trong Express	Route đường dẫn domain/admin để có thể vào được trang giao diện
3	<p>Tạo hàm xử lý dang_nhap</p> <p>Trong file routes/admin.js</p>	Có nhiệm vụ load trang đăng nhập khi người dùng gõ vào đường dẫn đã route phía trên
4	<p>Tạo hàm xử lý post_dang_nhap_admin</p> <p>Trong file routes/admin.js</p>	<p>Có nhiệm vụ nhận dữ liệu khi form đăng nhập được post lên từ người dùng sau khi họ nhập xong.</p> <p>Cho chuỗi mặc định để kiểm tra dữ liệu gửi lên có phù hợp hay không.</p>

5.5. Xây dựng trang danh sách slide banner trong admin:

- Yêu cầu: Thiết kế giao diện như hình:



- Hướng dẫn:

STT	Yêu cầu	Hướng dẫn
1	Thực hiện giao diện	Sử dụng HTML, CSS để tạo giao diện giống như hình đã cho
2	Tạo route trong Express	Route đường dẫn domain/admin/ danh-sach-slide-banner để có thể vào được trang giao diện
3	Tạo hàm xử lý <code>danh_sach_slide_banner</code> Trong file routes/admin.js	Có nhiệm vụ xử lý hiển thị trang khi người dùng gõ vào đường dẫn

BÀI 6: Client Session



Mục tiêu chính:

Áp dụng Client Session để xây dựng các trang cần đến session để lưu lại dữ liệu không mất khi qua trang khác ví dụ như: đăng nhập, giỏ hàng,...

6.1. Xây dựng trang đăng nhập với Client Session :

- Yêu cầu: Giao diện trang Đăng nhập website hoàn chỉnh:

– Thanh menu khi chưa đăng nhập:



– Form đăng nhập giống bài tập 5.1, lưu lý khi người dùng click vào menu Đăng nhập thì Popup form mới hiện ra

– Sau khi đăng nhập thành công giao diện menu sẽ như sau (dù người dùng có refresh trang cũng không mất trạng thái này vì thông tin đăng nhập đã lưu vào session):



- Hướng dẫn:

STT	Yêu cầu	Hướng dẫn
1	Tạo session – client chứa thông tin đăng nhập	Áp dụng kiến thức đã học tạo ra session dành cho thông tin đăng nhập Dùng lệnh <code>app.use()</code> để đưa session đó vào sử dụng, nhớ phải đưa thông tin cấu hình session đó vào.
2	Code xử lý đăng nhập sẽ lưu ở file <code>routes/nguoi_dung.js</code>	+ Xử lý thông tin post lên từ Form đăng nhập. Nếu hợp lệ thì lưu lại vào session – client + Lưu ý: hợp lệ ở đây là dùng một dữ liệu giả nào đó vì chúng ta chưa dùng đến CSDL, hoặc đọc từ một file txt thông tin người dùng (tùy các bạn chọn phương thức hợp lệ nào)
3	Hiển thị lại menu	Khi đăng nhập thành công phải cho hiển thị lại menu không còn menu đăng nhập nữa



6.2. Xây dựng trang đăng ký:

- Yêu cầu: Giao diện như sau:



Tên đăng nhập	<input type="text"/>
Mật khẩu	<input type="password"/>
Nhập lại mật khẩu	<input type="password"/>
Họ tên	<input type="text"/>
Ngày sinh	<input type="text" value="dd/mm/yyyy"/>
Email	<input type="text"/>
Giới tính:	<input type="text" value="Nam"/>
Avatar:	<input type="button" value="Chọn tệp"/> Không có tệp nào được chọn

- Hướng dẫn:

STT	Yêu cầu	Hướng dẫn
1	Tạo giao diện	Như hình đã cho
2	Tạo hàm xử lý trong file routes/nguoi_dung.js	Xử lý dữ liệu từ form gửi lên, sau đó nếu hợp lệ hết sẽ lưu vào session và đưa người dùng trở về


6.3. Thực hiện trang giỏ hàng:

- Yêu cầu: Xây dựng trang giỏ hàng có giao diện như sau:

STT	Mã video	Tên video	Hình	Đơn giá (dvd/blueray)	Số lượng (dvd/blueray)	Thành tiền
1		Iron Man 4		10000 đ/ 40000 đ	<input type="text" value="2"/> / <input type="text" value="3"/>	140000 đ
2		wolverine 3		10000 đ/ 40000 đ	<input type="text" value="1"/> / <input type="text" value="2"/>	90000 đ

Tổng cộng: 230000 đ

▪ Hướng dẫn:

STT	Yêu cầu	Hướng dẫn
1	Tạo giao diện	Sử dụng CSS, HTML để tạo giao diện như hình
2	Tạo nút đặt dvd tại trang chi tiết video	Xây dựng trang chi tiết video đơn giản với dữ liệu giả Tạo nút đặt dvd, lưu ý có 2 input số lượng riêng cho 2 loại là dvd thường và dvd blueray để người dùng nhập vào số lượng mà họ muốn đặt
3	Xử lý thao tác khi người dùng click vào nút đặt	Khi người dùng click vào nút đặt dvd thì gửi dữ liệu lên trang giỏ hàng xử lý Sau đó, lưu thông tin video được đặt DVD vào session -client
4	Hiển thị số lượng sản phẩm có trong giỏ hàng lên menu hình giỏ hàng	 Đặt luôn đường dẫn trên menu để người dùng click vào menu này sẽ vào trang thanh toán (có hiển thị danh sách giỏ hàng giống hình giao diện)
5	Xử lý cập nhật số lượng sản phẩm giỏ	Tạo trang thanh toán và sẽ chịu sự chi phối xử lý của hàm thanh_toan trong file



	hàng	<p>routes/video.js</p> <p>Gợi ý: sử dụng xử lý mảng để cập nhật thông tin trong session và lưu lại sau khi xử lý xong</p>
--	------	---

BÀI 7: Xây dựng web với CSDL MongoDB



Mục tiêu chính:

Xây dựng website dynamic với đầy đủ các chức năng cần sử dụng CSDL với CSDL là MongoDB.

7.1. Thực hiện load toàn bộ menu thật từ CSDL:

- Giao diện như hình bài tập 7.1:
- Vì menu sẽ dùng cho tất cả các trang, nên dữ liệu từ CSDL của menu sẽ được dùng lại nhiều lần => phải viết bằng Middleware để dễ dàng kế thừa lại.
- Cấu trúc file CSDL:

<input type="checkbox"/> _id	ObjectId("589c2ae18618e4a49795dcec")	ObjectId
<input checked="" type="checkbox"/> tieu_de	Đăng nhập	String
<input checked="" type="checkbox"/> alias	dang-nhap	String
<input checked="" type="checkbox"/> link		String
<input checked="" type="checkbox"/> trang_thai	1	Int32
<input checked="" type="checkbox"/> sap_xep	5	Int32
<input checked="" type="checkbox"/> da_dang_nhap	0	Int32

7.2. Hiện thị dữ liệu đối tác:

- Giao diện slide đối tác như hình:



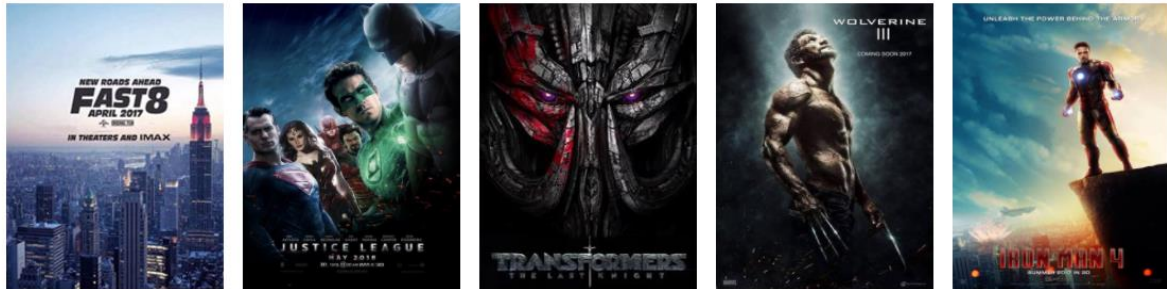
- Với đối tác chỉ hiển thị ở trang chủ nên viết trực tiếp trong route xử lý của trang chủ chứ không viết ngoài middleware ở bài 8.1
- Cấu trúc CSDL đối tác:

<input type="checkbox"/> _id	ObjectId("58942d92b9a27b4879e26c66")	ObjectId
<input checked="" type="checkbox"/> ten_doi_tac	Papaya studio	String
<input checked="" type="checkbox"/> hình	Papaya_studio.jpg	String

7.3. Hiện thị video nổi bật, tin tức nổi bật:


- Giao diện module video nổi bật ở trang chủ:

Video nổi bật trong tuần



- Giao diện module tin nổi bật:

Tin nổi bật




Tại sao Trương Vệ Kiện gắn liền với hình ảnh 'tóc gió thổi bay'?

ngày đăng: 3/1/2017

Phần trình diễn của Trương Vệ Kiện trong sự kiện hát mừng năm mới tại tỉnh Tứ Xuyên. Anh thể hiện loạt ca khúc hot như We Will Rock You,...

18 Comments 35 Likes




'Goblin' đã hết nhưng vẫn cuồng Gong Yoo, phải làm sao đây?

ngày đăng: 3/1/2017

Trước khi gây sốt với "Goblin" hay "Train to Busan", Gong Yoo đã cùng với Jeon Do Yeon viết nên một câu chuyện tình nặng trĩu những suy t...

18 Comments 35 Likes



KungFu Yoga – Cùng Thành Long đại náo cả Ấn Độ lẫn Ả-Rập

ngày đăng: 3/1/2017

Phim mới của ngôi sao võ thuật Hong Kong được đầu tư khá "khủng" với bối cảnh trải dài và số lượng diễn viên lớn, hứa hẹn những cảnh quay...

18 Comments 35 Likes

- Cấu trúc CSDL của tin tức, video:
- Cấu trúc CSDL tin tức:






<input type="checkbox"/> _id	ObjectId("589421b6b9a27b4879e2691a")	<input type="checkbox"/> ObjectId
<input type="checkbox"/> tieu_de	Tại sao Trương Vệ Kiện gắn liền với hình ảnh 'tóc gió thổi ...	<input type="checkbox"/> String
<input type="checkbox"/> noi_dung_tom_tat	Phần trình diễn của Trương Vệ Kiện trong sự kiện hát mùng...	<input type="checkbox"/> String
<input type="checkbox"/> noi_dung_chi_tiet	<div class="/"-conten/"ews newsquote"> Trương Vệ Kiện ...	<input type="checkbox"/> String
<input type="checkbox"/> hinh_dai_dien	3a10de6c9abf74f2731f6c97d92e5227_16300002a5dbfb8189...	<input type="checkbox"/> String
<input type="checkbox"/> noi_bat	true	<input type="checkbox"/> Boolean
<input type="checkbox"/> ngay_tao	1486102954	<input type="checkbox"/> Int32
<input type="checkbox"/> trang_thai	1	<input type="checkbox"/> Int32

▪ Cấu trúc CSDL video:

<input type="checkbox"/> _id	ObjectId("5881b297b9a27b4879e24188")	ObjectId
<input type="checkbox"/> ten_video	Fast And Furious 8	String
<input type="checkbox"/> hình	Fast-8-2017-watch-movie-full-free-hd-online-download-t...	String
<input type="checkbox"/> mo_ta	Đúng một ngày trước khi tung ra trailer đầu tiên, hãng Uni...	String
<input type="checkbox"/> link_video	fast_furious_8.mp4	String
<input type="checkbox"/> id_nguoi_dung	5881ae7db9a27b4879e2411a	String
<input type="checkbox"/> ngay_tao	1486103954	Int32
<input type="checkbox"/> trang_thai	1	Int32
<input type="checkbox"/> thoi_luong	126	Int32
<input type="checkbox"/> so_luot_down	5	Int32
<input type="checkbox"/> don_gia_dvd	10000	Int32
<input type="checkbox"/> don_gia_blueray	40000	Int32
<input type="checkbox"/> noi_bat	1	Int32

7.4. Các trang quản lý Video:

▪ Giao diện trang danh sách video admin:

Danh sách Video					
			Tìm kiếm <input type="text"/>		
STT	Tên video	Hình	file / link	Thao tác	
1	Avatar 2		avatar_2.mp4	Cập nhật	Xóa
2	Fast And Furious 8		fast_furious_8.mp4	Cập nhật	Xóa
3	Godzilla 2		https://www.youtube.com/embed/P2yx0hE3WJs	Cập nhật	Xóa
4	Guardian		https://www.youtube.com/embed/B16Bo47KS2g	Cập nhật	Xóa
5	Hell Boy		https://www.youtube.com/embed/QcYpGrjmdg4	Cập nhật	Xóa
<div> Đang hiển thị trang 1 trong 2 trang </div> <div> Trang trước 1 2 Trang sau </div>					

▪ Giao diện cập nhật, thêm video:

Thêm Video mới

Tên video

Mô tả

Mã HTML

B
I
U
 x_2
 x^2

Định dạng
Phông
Cỡ ...

body

File/Link

Thời lượng

Trạng thái

Nổi bật

Hình đại diện video:

Chọn tệp

Không có tệp nào được chọn

Lưu lại

- Giao diện cập nhật video:

Cập nhật video #58808cddb9a27b4879e23b89

Tên video

Avatar 2

Mô tả

Mã HTML

↶

↷

↺

↻

↵

↶

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

B

I

U

abε

x_2

x^2

☰

☷

☰

☷

☰

☷

☰

☷

☰

☷

☰

☷

Định dạng

Phông

Cỡ ...

Avatar là một bộ phim khoa học viễn tưởng của Mỹ năm 2009 do James Cameron viết kịch bản và đạo diễn, với sự tham gia của các diễn viên Sam Worthington, Zoe Saldana, Stephen Lang, Michelle

body

File/Link

avatar_2.mp4

Thời lượng

122

Trạng thái

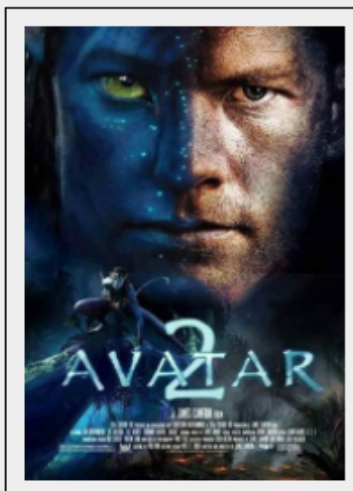
hiển thị

Nổi bật

Bình thường

Hình đại diện video:

Không có tệp nào được chọn



Lưu lại


- Cấu trúc CSDL của tin tức, video:
- Cấu trúc CSDL video:

<input type="text"/>	_id	ObjectId("5881b297b9a27b4879e24188")	ObjectId
<input type="text"/>	ten_video	Fast And Furious 8	String
<input type="text"/>	hinh	Fast-8-2017-watch-movie-full-free-hd-online-download-t...	String
<input type="text"/>	mo_ta	Đúng một ngày trước khi tung ra trailer đầu tiên, hãng Uni...	String
<input type="text"/>	link_video	fast_furious_8.mp4	String
<input type="text"/>	id_nguoi_dung	5881ae7db9a27b4879e2411a	String
<input type="text"/>	ngay_tao	1486103954	Int32
<input type="text"/>	trang_thai	1	Int32
<input type="text"/>	thoi_luong	126	Int32
<input type="text"/>	so_luot_down	5	Int32
<input type="text"/>	don_gia_dvd	10000	Int32
<input type="text"/>	don_gia_bluray	40000	Int32
<input type="text"/>	noi_bat	1	Int32

7.5. Trang danh sách người dùng:

- Giao diện:


Danh sách người dùng



hungnguyen

Hùng Nguyễn


Online



linhnguyen

Linh Nguyễn


Offline



nhattran

Nhật Trần


Offline



myphuong

Lê Thị Mỹ Phương


Offline



ngocmy

Dương Ngọc My


Offline



minhvuong

Mạc Minh Vương

Offline



thuydung

Võ Thùy Dung

Offline

- Cấu trúc CSDL người dùng:

<input type="text"/>	_id	ObjectId("5881ae7db9a27b4879e2411a")	ObjectId
<input type="text"/>	ho_ten	Hùng Nguyễn	String
<input type="text"/>	ngay_dang_ky	2016-12-5	String
<input type="text"/>	tai_khoan	hungnguyen	String
<input type="text"/>	mat_khau	123456	String
<input type="text"/>	email	hungnguyenxuan118@gmail.com	String
<input type="text"/>	gioi_tinh	0	Int32
<input type="text"/>	avatar	nam1.jpg	String
<input type="text"/>	id_quyen_han	58a2ca401d2c5cd581eadfcf	String
<input type="text"/>	online	1	Int32
<input type="text"/>	thoi_gian_dang_nhap	1496028195	Int32

7.6. Bình luận trong tin tức và video:

- Giao diện:



Bình luận

Hùng Nguyễn
Bài viết rất hay. Mình rất thích!

Linh Nguyễn
Mong sẽ có nhiều bài viết hay giống vậy nữa nhé ad!

Nhật Trần
thử lần nữa nè

Gửi bình luận

▪ Cấu trúc CSDL bình luận:

<input type="text"/> _id	ObjectId("5899392b8618e4a49794d4cb")	ObjectId
<input type="text"/> noi_dung	Bài viết rất hay. Mình rất thích!	String
<input type="text"/> id_tin_tuc	589422afb9a27b4879e26963	String
<input type="text"/> id_video		String
<input type="text"/> id_nguoi_dung	5881ae7db9a27b4879e2411a	String
<input type="text"/> ngay_tao	1486105555	Int32

BÀI 8: Socket IO










Mục tiêu chính:



Xây dựng website với các chức năng Real-time thông qua thư viện Socket IO trong NodeJS

8.1. Real time trạng thái người dùng đăng nhập vào website:

- Giao diện trang danh sách người dùng admin cùng trạng thái:

Danh sách người dùng





 hungnguyen Hùng Nguyễn ● Online	 linhnguyen Linh Nguyễn ● Offline	 nhattran Nhật Trần ● Offline	 myphuong Lê Thị Mỹ Phương ● Offline
 ngocmy Dương Ngọc My ● Offline	 minhvuong Mạc Minh Vương ● Offline	 thuydung Võ Thùy Dung ● Offline	

- Các chấm tròn có màu xanh lá  là đăng online, còn màu đỏ  là đăng offline
- Hướng dẫn:

STT	Yêu cầu	Hướng dẫn
1	Chèn thư viện socket.IO	Khai báo thư viện socketIO Dùng lệnh app.set để gọi thư viện SocketIO vào sử dụng
2	Tạo Emitter phía Server gọi Listener Client khi đăng nhập thành công	Dùng thư viện socketIO tạo Emitter gọi Client cập nhật danh sách người dùng cùng trạng thái
3	Xây dựng Listener thay đổi trạng thái	Listener phía Client để thao tác thay đổi trạng thái người dùng.

8.2. Real time số lượng người dùng đăng online ở trang thống kê:

- Giao diện:

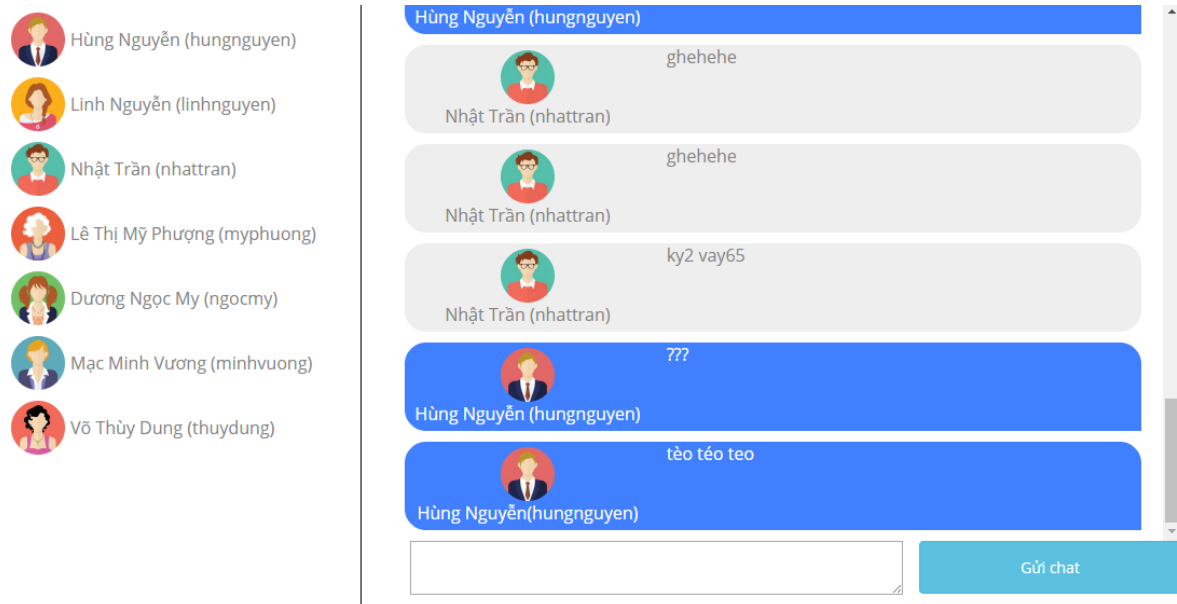
 67 TỔNG LƯỢT DOWNLOAD	 7 TỔNG SỐ THÀNH VIÊN	 1 SỐ THÀNH VIÊN ĐANG ONLINE	 10 TỔNG SỐ VIDEO
---	--	---	---

- Với ô màu đen chứa số lượng người dùng đăng thực sự online.

- Khi có người dùng khác đăng nhập, số lượng này sẽ tăng lên, hay khi người dùng đăng xuất thì số lượng này sẽ tự giảm xuống

8.3. Viết tool chat chung với tất cả thành viên:

- Giao diện tool chat như sau:



- CSDL chat all có cấu trúc:

<code>_id</code>	<code>ObjectId("592d16884ed36a0ded5fd2b9")</code>	<code>ObjectId</code>
<code>id_nguoi_dung</code>	<code>5881ae7db9a27b4879e2411a</code>	<code>String</code>
<code>noi_dung</code>	<code>test</code>	<code>String</code>
<code>ngay_tao</code>	<code>132300004040</code>	<code>Int64</code>

- Hướng dẫn code:

STT	Yêu cầu	Hướng dẫn
1	Tạo một trang <code>pages/trang_chat.ejs</code>	Với thiết kế giao diện giống như hình đã cho
2	Tạo listener và emitter phía server	Listener phía server có nhiệm vụ nhận dữ liệu từ client gửi qua và lưu vào CSDL sau đó trả về giá trị đã lưu Emitter phía server có nhiệm vụ gửi lại giá trị sau khi lưu và được gọi khi listener xử lý xong
3	Tạo listener và emitter phía client	Listener phía client nhận dữ liệu sau khi lưu của phía Server gửi về và hiển thị lên trên khung chat Emitter phía client gửi dữ liệu form chat khi người dùng chat về phía Server

BÀI 9: RESTFul API



Mục tiêu chính:

Giúp học viên hoàn thành một ứng dụng web và đưa lên mạng

9.1. Làm service API trả về danh sách video lazy load, và tin tức lazy load:

Thực hiện api với phương thức GET:

- Xây dựng 2 hàm API: hàm lazy_load_video, hàm lazy_load_tin_tuc
- Tham số truyền vào hàm là trang hiện tại.
- Dữ liệu trả về là danh sách video hoặc tin tức kế tiếp của trang hiện tại.

9.2. Làm service API cho bình luận:

Thực hiện api với phương thức GET:

- Xây dựng hàm API với tên danh_sach_binh_luan_video
- Tham số truyền vào là id_video
- Nếu id_video có giá trị rỗng, trả về danh sách toàn bộ bình luận của video, còn id_video có giá trị thì chỉ lấy tất cả danh sách bình luận của video này. Lưu ý: danh sách bình luận phải được sắp xếp theo thời gian tăng dần

Thực hiện api với phương thức POST:

- Xây dựng hàm API với tên them_binh_luan
- Tham số truyền vào gồm 2 tham số: id_video hoặc id_tin_tuc, loai_binh_luan
- Nếu loai_binh_luan có giá trị là video thì sẽ lưu bình luận có thuộc tính id_video có giá trị là id_video được truyền vào
- Nếu loai_binh_luan có giá trị là tin_tuc thì bình luận sẽ được lưu có thuộc tính id_tin_tuc với giá trị là id_tin_tuc được truyền vào hàm

CSDL của bình luận như sau:

<input type="checkbox"/> _id	ObjectId("5899392b8618e4a49794d4cb")	ObjectId
<input type="checkbox"/> noi_dung	Bài viết rất hay. Mình rất thích!	String
<input type="checkbox"/> id_tin_tuc	589422afb9a27b4879e26963	String
<input type="checkbox"/> id_video		String
<input type="checkbox"/> id_nguoi_dung	5881ae7db9a27b4879e2411a	String
<input type="checkbox"/> ngay_tao	1486105555	Int32



9.3. Làm service API cho một số thống kê:

Với việc xây dựng các API thống kê thì đa số các phương thức thường được sử dụng là phương thức GET:

- Xây dựng API thống kê tổng số lượt download của tất cả video.
 - Xây dựng API đếm số lượng dòng dữ liệu của một collection được truyền tên vào hàm.
- Ví dụ: nếu tham số truyền vào là collection có tên video, thì hàm sẽ đếm xem toàn bộ video có trong CSDL là bao nhiêu.

BÀI 10: Upload website lên Heroku



Mục tiêu chính:

Giúp học viên hoàn thành một ứng dụng web và đưa lên mạng

10.1. Upload website lên heroku:

Phần 1: Tạo tài khoản trên heroku

- ✓ Truy cập vào website <https://www.heroku.com/> để tạo tài khoản heroku:

Phần 2: Tạo tài khoản trên githud:

- ✓ Truy cập vào website <https://github.com/> để tạo tài khoản upload source code.

Phần 3: Cài đặt cấu hình Heroku

- ✓ Upload source code lên githud
- ✓ Tạo app trên heroku
- ✓ Cấu hình app trên heroku trở về source githud
- ✓ Deloy ứng dụng.



Mục Lục

BÀI 1: Tổng quan về NodeJS	2
1.1. Làm quen với NodeJS	2
1.2. Làm quen với NodeJS	2
BÀI 2: Tạo server web bằng Http.....	3
2.1. Tạo server web với Http, lắng nghe từ port 8080:	3
2.2. Tính tổng 2 số, với giá trị 2 số lấy từ đường dẫn:	3
2.3. In bảng cửu chương từ biến post:	3
BÀI 3: Module và Events	5
3.1. Xây dựng module xử lý định dạng chữ.....	5
3.2. Xây dựng module dạng class đối tượng trong NodeJS:	6
3.3. Tạo sự kiện với EventEmitter:	7
3.4. Tạo sự kiện với EventEmitter, và response trong Event mới tạo chứ không response trực tiếp từ Http server:	8
BÀI 4: Buffer và Stream	9
4.1. Tạo ứng dụng tính tiền khi bán ngoại tệ và lưu lại danh sách giao dịch:	9
4.2. Xây dựng trang xem lại lịch sử giao dịch đã bán:.....	10
4.3. Xây dựng trang Cập nhật tỷ giá bán:	11
BÀI 5: Express Framework	12
5.1. Tạo Form và thực hiện đăng nhập người dùng:	12
5.2. Tạo Middleware đơn giản kiểm tra đăng nhập:.....	13
5.3. Tạo Middleware load danh sách menu lên trang web:	13
5.4. Tạo Middleware hiển thị lỗi khi không tìm thấy trang:.....	14
BÀI 6: EJS template	15
6.1. Tạo trang chủ như template:.....	15
6.2. Tạo trang danh sách video:.....	16



6.3.	Xây dựng trang chi tiết video:	18
6.4.	Giao diện login admin:.....	19
6.5.	Xây dựng trang danh sách slide banner trong admin:	20
BÀI 7: Client Session		21
7.1.	Xây dựng trang đăng nhập với Client Session :	21
7.2.	Xây dựng trang đăng ký:	22
7.3.	Thực hiện trang giỏ hàng:	22
BÀI 8: Xây dựng web với CSDL MongoDB.....		25
8.1.	Thực hiện load toàn bộ menu thật từ CSDL:	25
8.2.	Hiển thị dữ liệu đối tác:.....	25
8.3.	Hiển thị video nổi bật, tin tức nổi bật:	25
8.4.	Các trang quản lý Video:.....	27
8.5.	Trang danh sách người dùng:.....	30
8.6.	Bình luận trong tin tức và video:	30
BÀI 9: Socket IO.....		32
9.1.	Real time trạng thái người dùng đăng đăng nhập vào website:	32
9.2.	Real time số lượng người dùng đăng online ở trang thống kê:	32
9.3.	Viết tool chat chung với tất cả thành viên:	33
BÀI 10: RESTFul API.....		34
10.1.	Làm service API trả về danh sách video lazy load, và tin tức lazy load:	34
10.2.	Làm service API cho bình luận:	34
10.3.	Làm service API cho một số thống kê:	35
BÀI 11: Upload website lên Heroku.....		36