KHỞI TẠO DỰ ÁN - EXPRESSJS - MONGODB

1. Khởi tạo Node Package Manager (NPM) npm init -y
2. Tạo file index.js
3. Khởi tạo dự án Expressjs npm install express
4. Chạy thử dự án demo Hello World tại [link](https://expressjs.com/en/starter/hello-world.html)
5. Cài đặt package để khi save tự reload lại trang web [link](https://www.npmjs.com/package/nodemon)
   * "start": "nodemon –-inspect index.js",
6. Commit dự án lên github (lưu ý cần phải có gitignore)
7. Chia router (router admin trước, tạm thời là dashboard) → commit github
8. Chia controller (controller admin trước, tạm thời là dashboard) → commit github
9. Cài đặt package .env https://www.npmjs.com/package/dotenv
10. Cấu hình file .env (cấu hình port trước) → commit github (Lưu ý file này không được commit lên)
11. Cài đặt template engines
    * pug [link](https://www.npmjs.com/package/pug)
    * plugin với expressjs [link](https://expressjs.com/en/guide/using-template-engines.html)
    * Lưu ý phải cài cả 2 link trên
12. Cài đặt static file (là những file được public ra FrontEnd) [link](https://expressjs.com/en/starter/static-files.html)
13. Cấu trúc thư mục Views
14. Cài đặt mongoose để kết nối với MongoDB
    * Thư viện mongoose [link](https://www.npmjs.com/package/mongoose) (mongoose)
    * Thư viện hỗ trợ tạo slug với mongoose [link](https://www.npmjs.com/package/mongoose-slug-updater) (mongoose-slug-updater**)**
    * có thể dùng thư viện này để tự tạo slug riêng [link](https://www.npmjs.com/package/unidecode) (unidecode)
    * Viết file kết nối mongoose với MongoDB
15. Lưu ý có thể cài đặt thêm SCSS nếu xây dựng dự án bằng scss, và chỉ cần lệnh render từ SCSS ra CSS
    * Câu lệnh cài đặt (dùng phiên bản này để có thể dùng câu lệnh import) npm instal sass@1.77.8
    * "compile-scss": "sass --watch assets/scss:assets/css"
16. Lưu ý có thể render từ PUG ra HTML (Nếu cần)
    * "compile-pug": "pug -w ./template -o ./ -P"
17. Thư viện chạy nhiều câu lệnh NPM cùng một lúc [link](https://www.npmjs.com/package/npm-run-all) (npm-run-all)

XÂY DỰNG TRANG ADMIN

1. Quản Lý Sản Phẩm
   * Giao diện trang Quản Lý Sản Phẩm
   * Router /admin/products
   * Controller (ban đầu thì sẽ render ra giao diện tĩnh để xem kết nối được chưa)
   * Xây dựng Database products
   * Controller lấy dữ liệu và render ra giao diện Quản Lý Sản Phẩm
   * Tính năng
     + Bộ Lọc Trạng Thái
       1. URL : /admin/products?status=type (GET)
       2. Các type
          1. active: hoạt động
          2. inactive: dừng hoạt động
          3. pending: đang chờ xử lý
     + Tìm kiếm theo keyword
       1. URL: /admin/products?keyword=từ khóa tìm kiếm (GET)
       2. Tìm kiếm căn bản, không phân biệt hoa thường nhưng không thể phân biệt có dấu hay không có dấu
          1. Phải sử dụng regex để tìm kiếm, flag “i” để tìm kiếm không phân biệt hoa thường
       3. Tìm kiếm nâng cao dựa vào slug sử dụng thêm thư viện [link](https://www.npmjs.com/package/unidecode) (unidecode)
     + Phân trang (pagination)
       1. URL: /admin/products?page=số trang (GET)
       2. Tính toán
          1. Lấy trang hiện tại
          2. Lấy tổng số sản phẩm có trong database
          3. Chỉ định giới hạn sản phẩm của một trang
          4. Di chuyển đến sản phẩm trang hiện tại = (Trang hiện tại – 1) \* Giới hạn sản phẩm 1 trang
          5. Tổng số trang = ceil(Tổng số sả phẩm có trong database / G.hạn sản phẩm 1 trang)
     + Thay đổi trạng thái sản phẩm
       1. URL: /admin/products/change-status/:id/:status (PATCH)
       2. Bởi vì Form chỉ gửi được method GET / POST nên cần dùng thêm thư viện
          1. thư viện ghi đè method [link](https://www.npmjs.com/package/method-override) (method-override)
       3. Thông báo khi thay đổi trạng thái sản phẩm thành công hoặc thất bại
          1. thư viện hỗ trợ thông báo alert [link](https://www.npmjs.com/package/express-flash) (express-flash)

thư viện đi kèm [link](https://www.npmjs.com/package/cookie-parser) (cookie-parser)

thư viện đi kèm [link](https://www.npmjs.com/package/express-session) (express-session)

giải thích vì sao phải dùng thêm 2 thư viện đi kèm [link](https://stackoverflow.com/questions/25550819/error-most-middleware-like-bodyparser-is-no-longer-bundled-with-express)

* + - 1. Lưu ý thêm tính năng check xem trạng thái gửi lên có nằm trong danh sách trạng thái không nếu không thì không cho cập nhật
      2. Lưu userID và ngày thay đổi trạng thái
    - Xóa mềm sản phẩm
      1. URL: /admin/products/delete-soft/:id (DELETE)
      2. Bởi vì Form chỉ gửi được method GET / POST nên cần dùng thêm thư viện
         1. thư viện ghi đè method [link](https://www.npmjs.com/package/method-override) (method-override)
      3. Thông báo khi thay đổi trạng thái sản phẩm thành công hoặc thất bại
         1. thư viện hỗ trợ thông báo alert [link](https://www.npmjs.com/package/express-flash) (express-flash)

thư viện đi kèm [link](https://www.npmjs.com/package/cookie-parser) (cookie-parser)

thư viện đi kèm [link](https://www.npmjs.com/package/express-session) (express-session)

giải thích vì sao phải dùng thêm 2 thư viện đi kèm [link](https://stackoverflow.com/questions/25550819/error-most-middleware-like-bodyparser-is-no-longer-bundled-with-express)

* + - Thay đổi trạng thái / vị trí / xóa nhiều sản phẩm
      1. Tạo checkbox
      2. Tạo hộp lựa chọn (như là cái select): status, position, delete,…
      3. URL: /admin/products/change-multi (PATCH)
      4. Submit qua form
         1. Kiểu thay đổi: status, position, delete
         2. Mảng ID sản phẩm
      5. Thông báo sau khi thay đổi thành công
    - Tạo mới sản phẩm
      1. Giao diện tạo mới sản phẩm
      2. Đường dẫn
         1. /admin/products/create (GET) đường dẫn vào trang tạo mới
         2. /admin/products/create POST) đường dẫn khi tạo sản phẩm
         3. Tuy đường dẫn giống nhau như khác phương thức thì nó khác nhau nha
      3. Khi tạo mới sản phẩm cần submit qua form nên ta cần các thư viện
         1. Thư viện để req.body nhận được data [link](https://www.npmjs.com/package/body-parser) (body-parser)
      4. Upload hình ảnh [link](https://www.npmjs.com/package/multer) (multer). Đọc kĩ hướng dẫn bên dưới này
         1. Lưu ý thêm trường thuộc tính này vào form

A screenshot of a computer

Description automatically generated

* + - * 1. Lưu ý: khi uploads hình ảnh, ban đầu nếu chưa upload lên Cloudinary hoặc một ứng dụng Cloud nào đó thì ta cần upload vào một thư mục. Sau khi upload vào thư mục thì trường dữ liệu chứa ảnh sẽ lưu đường dẫn đến thư mục đó. Nhờ vậy khi render ra giao diện thì nó sẽ trỏ vào thư mục chưa file để lấy ra ảnh
        2. Cách preview hình ảnh lên giao diện trước khi upload [link](https://stackoverflow.com/questions/4459379/preview-an-image-before-it-is-uploaded)
        3. Cách xóa giá trị trong thẻ input type file [link](https://stackoverflow.com/questions/3144419/how-do-i-remove-a-file-from-the-filelist)
      1. Liên kết cloudinary
         1. Trang Cloudinary [link](https://cloudinary.com/)
         2. Hướng dẫn upload ảnh lên Cloudinary [link](https://cloudinary.com/blog/node_js_file_upload_to_a_local_server_or_to_the_cloud)
         3. API reference [link](https://cloudinary.com/documentation/image_upload_api_reference)
         4. Upload file, audio… [link](https://support.cloudinary.com/hc/en-us/articles/115003361672-Can-I-upload-audio-files-to-my-Cloudinary-account)
      2. Chèn text editor Tiny-mce cho phần mô tả sản phẩm
         1. Tích hợp tiny-mce vào ứng dụng expressjs [link](https://www.tiny.cloud/docs/tinymce/latest/expressjs-pm/)
         2. Image và File option (chèn ảnh, … ) [link](https://www.tiny.cloud/docs/tinymce/latest/file-image-upload/#interactiveexample)
         3. Image plugin [link](https://www.tiny.cloud/docs/tinymce/latest/image/)
         4. Image edit [link](https://www.tiny.cloud/docs/tinymce/latest/editimage/)
         5. Plugin for tiny-mce [link](https://www.tiny.cloud/docs/tinymce/latest/plugins/)
      3. Lựa chọn Danh Mục sản phẩm (Có thể làm sau)
      4. Middleware Validate khi tạo sản phẩm
      5. Thông báo khi tạo sản phẩm thành công
    - Chỉnh sửa sản phẩm
      1. Giao diện chỉnh sửa sản phẩm
      2. Đường dẫn
         1. /admin/products/edit/:id (GET) đường dẫn vào trang chỉnh sửa
         2. /admin/products/edit/:id (PATCH) đường dẫn khi tạo chỉnh sửa
         3. Tuy đường dẫn giống nhau như khác phương thức thì nó khác nhau nha
      3. Khi chỉnh sửa sản phẩm cần submit qua form nên ta cần các thư viện (Không cài lại nếu đã cài trước đó)
         1. Thư viện để req.body nhận được data [link](https://www.npmjs.com/package/body-parser) (body-parser)
      4. Upload hình ảnh [link](https://www.npmjs.com/package/multer) (multer)
      5. Lựa chọn Danh Mục sản phẩm
      6. Middleware Validate khi chỉnh sửa sản phẩm
      7. Lưu ý khi chỉnh sửa sản phẩm mà ta không cập nhật ảnh mới thì khi submit lên sẽ không có ảnh đi kèm. Lúc này ta nên loại bỏ trường hình ảnh để tránh cập nhật sai
    - Chi tiết sản phẩm
      1. URL: /admin/products/detail/:id (GET)
    - Sắp xếp sản phẩm theo tiêu chí
      1. Tiêu chí
         1. Vị trí: Tăng dần, giảm dần
         2. Chữ cái: Tăng dần, giảm dần
      2. URL: /admin/products?sortKey=\_\_\_&sortValue=\_\_\_ (GET)
      3. Lưu ý khi lấy sản phẩm ra dùng phương thức sort của database là được

MULTER – UPLOADS FILE

Thư viện: [multer](https://www.npmjs.com/package/multer)

1. Cách Upload Hình Ảnh Dưới Local
2. A screenshot of a computer

   Description automatically generatedA screenshot of a computer

   Description automatically generatedĐịnh nghĩa: Upload hình ảnh dưới local là sẽ upload file ảnh vào một thư mục (static file) để sau đó ta sẽ lưu đường dẫn ảnh vào trường dữ liệu trong database

Hình ảnh minh họa lưu vào local và trỏ vào thư mục

1. A computer screen with text and images

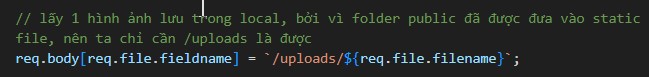
   Description automatically generatedViết hàm Disk Storage để lưu vào thư mục mình muốn lưu
2. A screen shot of a computer program

   Description automatically generatedGọi và sử dụng multer và diskStorage như một middleware

A computer screen with text

Description automatically generatedLưu ý “thumbnail” là tên trường sẽ lưu đường dẫn trong database

Khi dùng multer upload single thì sẽ có req.file

1. Xử lý lưu đường dẫn vào database
2. Cách upload nhiều ảnh (một mảng ảnh) dưới local
3. Trường lưu trữ ảnh trong database phải là một mảng

A screen shot of a computer program

Description automatically generated

1. Sử dụng lại bước (2) của upload một ảnh. Vì hiện tại đang upload dưới local lên cần một nơi lưu trữ ảnh
2. Định nghĩa middleware upload nhiều ảnh

A screen shot of a computer

Description automatically generated

Trường upload lên có tên là thumbnail và upload tối đa 3 ảnh

Lưu ý số lượng ảnh được upload lên có thể nhiều hơn

1. Xử lý đường dẫn vào database

A computer screen with text and images

Description automatically generated

Khi upload nhiều ảnh thì ta sẽ có req.files

1. Giải thích vì sao lại lưu là ‘/uploads/tên\_file\_ảnh’
   1. Bởi vì khi dùng static file (folder public) mặc định các file trong folder public đã tự trỏ được đến folder public vì vậy khi render link ảnh vào thẻ img <img src=”link\_ảnh”/> nó đã mặc định có là <img src=”public/link\_ảnh”/> nên ta chỉ cần lưu từ /uploads/….

**CLOUDINARY** lưu ý phải kết hợp với multer

|  |  |
| --- | --- |
| Trang chủ cloudinary | [link](https://cloudinary.com/) |
| Hướng dẫn upload ảnh lên cloudinary | [link](https://cloudinary.com/blog/node_js_file_upload_to_a_local_server_or_to_the_cloud) |
| API reference | [link](https://cloudinary.com/documentation/image_upload_api_reference) |
| Upload file, audio, … | [link](https://support.cloudinary.com/hc/en-us/articles/115003361672-Can-I-upload-audio-files-to-my-Cloudinary-account) |

**Bước 1:** Login vào trang chủ và cài đặt package cloudinary

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated**Bước 2:** Config API Cloudinary

Coppy đoạn code config để cấu hình

A screenshot of a computer

Description automatically generated

Vào đây để lấy API Keys. Lưu ý các API Keys cần được đưa vào file .env

A screen shot of a computer program

Description automatically generated

Một số lưu ý khi upload

* Đối với upload một ảnh thì multer sẽ trả ra req.file nên ta chỉ cần chọc thẳng vào
* Nhưng đối với upload nhiều ảnh thì multer sẽ trả ra req.files và đó là một mảng nên ta cần chú ý duyệt từng phần tử trong mảng

TINY-MCE sử dụng với textarea

1. A screenshot of a computer program

   Description automatically generatedKhi tích hợp thì cần import thêm thư viện path

**A screenshot of a computer

Description automatically generated**A screen shot of a computer code

Description automatically generatedThư viện path này có sẵn nên chỉ cần require để hỗ trợ

khi tích hợp tiny-mce thì phải tích hợp thêm dòng này là base tiny-mce

Cách kiểm thử: Lên URL thử đường dẫn này. Nếu như hình ảnh thì thành công

Lưu ý: Sau khi tích hợp thì nhớ chạy lại chương trình thì nó mới load lên được

|  |  |
| --- | --- |
| Tích hợp Tiny-mce với Expressjs | [link](https://www.tiny.cloud/docs/tinymce/latest/expressjs-pm/) |
| Chèn image | [link](https://www.tiny.cloud/docs/tinymce/latest/file-image-upload/#interactiveexample) |
| Image plugin | [link](https://www.tiny.cloud/docs/tinymce/latest/image/) |
| Image edit | [link](https://www.tiny.cloud/docs/tinymce/latest/editimage/) |
| Plugin for Tiny-mce | [link](https://www.tiny.cloud/docs/tinymce/latest/plugins/) |
| Placeholder cho tiny-mce | [link](https://www.tiny.cloud/blog/tinymce-placeholder-text/) |
| Tiny mce command | [link](https://www.tiny.cloud/docs/tinymce/latest/editor-command-identifiers/) |

**THẺ INPUT**

|  |  |
| --- | --- |
| Sử dụng javascript để thay đổi text placeholder | [link](https://stackoverflow.com/questions/13506481/change-placeholder-text) |
| Preivew hình ảnh trước khi upload | [link](https://stackoverflow.com/questions/4459379/preview-an-image-before-it-is-uploaded) |
| Remove background-color khi sử dụng autocomplete | [link](https://stackoverflow.com/questions/2781549/removing-input-background-colour-for-chrome-autocomplete) |
| Lắng nghe sự kiện thay đổi của thẻ input type file | [link](https://stackoverflow.com/questions/3528359/html-input-type-file-file-selection-event) |
| Sử dụng javascript check xem một element có chứa class đang tìm | [link](https://stackoverflow.com/questions/5898656/check-if-an-element-contains-a-class-in-javascript) |
|  |  |
|  |  |

Dùng thẻ label for tới cái id của thẻ input type file thì chỉ cần kích vào label nó sẽ mở thẻ input type file