

MERN stack - Shopping Online

Admin - listproduct, addproduct,
updateproduct, deleteproduct

Trang Hồng Sơn

tranghongson@gmail.com
tranghongson@vanlanguni.vn

Outline

1. Functionals

- 1.1 Admin - listproduct
- 1.2 Admin - addproduct
- 1.3 Admin - updateproduct
- 1.4 Admin - deleteproduct

Outline

1. Functionals

- 1.1 Admin - listproduct
- 1.2 Admin - addproduct
- 1.3 Admin - updateproduct
- 1.4 Admin - deleteproduct

Server

- Create 'server/models/ProductDAO.js' file

```
1 require('../utils/MongooseUtil');
2 const Models = require('../Models');
3
4 const ProductDAO = {
5   async selectAll() {
6     const query = {};
7     const products = await Models.Product.find(query).exec();
8     return products;
9   }
10 };
11 module.exports = ProductDAO;
```

Server

- Update 'server/api/admin.js' file

```
1  ...
2  // daos
3  const ProductDAO = require('../models/ProductDAO');
4  ...
5  // product
6  router.get('/products', JwtUtil.checkToken, async function (req, res) {
7    // get data
8    var products = await ProductDAO.selectAll();
9    // pagination
10   const sizePage = 4;
11   const noPages = Math.ceil(products.length / sizePage);
12   var curPage = 1;
13   if (req.query.page) curPage = parseInt(req.query.page); // /products?page=
14   xxx
15   const offset = (curPage - 1) * sizePage;
16   products = products.slice(offset, offset + sizePage);
17   // return
18   const result = { products: products, noPages: noPages, curPage: curPage };
19   res.json(result);
20 });
...

```

- Postman: (GET) <http://localhost:3000/api/admin/products?page=1>

- Headers: "x-access-token": <token>

Client-admin





React App

localhost:3001/admin/product

HOME CATEGORY PRODUCT ORDER CUSTOMER

Hello admin | Logout

PRODUCT LIST

ID	Name	Price	Creation date	Category	Image
6288bb8505f9bb7c9f45cc18	iPad Air	10	5/21/2022, 5:14:29 PM	iPad	
6288bd295ddf350614f2a916	iPad Pro	30	5/21/2022, 5:21:29 PM	iPad	
6288bc0940a1d1225e22805d	iPad Mini	20	5/21/2022, 5:16:41 PM	iPad	
6288bdda3afd8004d22ad823	iPhone 11 Pro Max	40	5/21/2022, 5:24:26 PM	iPhone	

1 | 2 | 3

PRODUCT DETAIL

ID

6288bb8505f9bb7c9f45

Name

iPad Air

Price

10

Image

Choose File

No file chosen


Category

iPad

ADD NEW

UPDATE

DELETE



Outline

1. Functionals

- 1.1 Admin - listproduct
- 1.2 Admin - addproduct
- 1.3 Admin - updateproduct
- 1.4 Admin - deleteproduct

Server

- Update 'server/models/CategoryDAO.js' file

```
1  ...
2  const CategoryDAO = {
3    ...,
4    async selectById(_id) {
5      const category = await Models.Category.findById(_id).exec();
6      return category;
7    }
8  };
9  ...
```

- Update 'server/models/ProductDAO.js' file

```
1  ...
2  const ProductDAO = {
3    ...,
4    async insert(product) {
5      const mongoose = require('mongoose');
6      product._id = new mongoose.Types.ObjectId();
7      const result = await Models.Product.create(product);
8      return result;
9    }
10 };
11 ...
```


Server

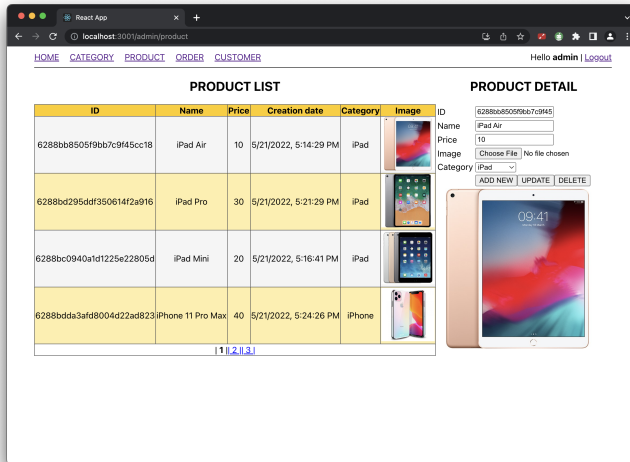
- Update 'server/api/admin.js' file

```
1  ...
2  // product
3  router.post('/products', JwtUtil.checkToken, async function (req, res) {
4      const name = req.body.name;
5      const price = req.body.price;
6      const cid = req.body.category;
7      const image = req.body.image;
8      const now = new Date().getTime(); // milliseconds
9      const category = await CategoryDAO.selectByID(cid);
10     const product = { name: name, price: price, image: image, cdate: now,
11                       category: category };
12     const result = await ProductDAO.insert(product);
13     res.json(result);
14     ...
```

- Postman: (POST) http://localhost:3000/api/admin/products

- Headers: "x-access-token": <token>
- Body (raw+JSON): { "name": "iWatch S7", "price": 130, "category": "644279d8146cdb5d63250036", "image": "/9j/4AAQSkZJRgABAQEAAAAAAAAAAD/..." }

Client-admin



Outline

1. Functionals

- 1.1 Admin - listproduct
- 1.2 Admin - addproduct
- 1.3 Admin - updateproduct
- 1.4 Admin - deleteproduct

Server

- Update 'server/models/ProductDAO.js' file

```
1  ...
2  const ProductDAO = {
3    ...,
4    async selectByID(_id) {
5      const product = await Models.Product.findById(_id).exec();
6      return product;
7    },
8    async update(product) {
9      const newvalues = { name: product.name, price: product.price, image:
10        product.image, category: product.category };
11      const result = await Models.Product.findByIdAndUpdate(product._id,
12        newvalues, { new: true });
13      return result;
14    }
15  };
16  ...
```

Server

- Update 'server/api/admin.js' file

```
1  ...
2  // product
3  router.put('/products', JwtUtil.checkToken, async function (req, res) {
4      const _id = req.body.id;
5      const name = req.body.name;
6      const price = req.body.price;
7      const cid = req.body.category;
8      const image = req.body.image;
9      const now = new Date().getTime(); // milliseconds
10     const category = await CategoryDAO.selectByID(cid);
11     const product = { _id: _id, name: name, price: price, image: image, cdate:
        now, category: category };
12     const result = await ProductDAO.update(product);
13     res.json(result);
14 });
15 ...
```

- Postman: (PUT) <http://localhost:3000/api/admin/products/<id>>

- Headers: "x-access-token": <token>
- Body (raw+JSON): { "name": "iWatch S7", "price": 130, "category": "644279d8146cdb5d63250036", "image": "/9j/4AAQSkZJRgABAQEAAAAAAAAAD/..." }

Client-admin





React App

localhost:3001/admin/product

HOME CATEGORY PRODUCT ORDER CUSTOMER

Hello admin | Logout

PRODUCT LIST

ID	Name	Price	Creation date	Category	Image
6288bb8505f9bb7c9f45cc18	iPad Air	10	5/21/2022, 5:14:29 PM	iPad	
6288bd295ddf350614f2a916	iPad Pro	30	5/21/2022, 5:21:29 PM	iPad	
6288bc0940a1d1225e22805d	iPad Mini	20	5/21/2022, 5:16:41 PM	iPad	
6288bdda3afd8004d22ad823	iPhone 11 Pro Max	40	5/21/2022, 5:24:26 PM	iPhone	

1 | 2 | 3

PRODUCT DETAIL

ID

6288bb8505f9bb7c9f45

Name

iPad Air

Price

10

Image

Choose File

No file chosen


Category

iPad

ADD NEW

UPDATE

DELETE



Outline

1. Functionals

- 1.1 Admin - listproduct
- 1.2 Admin - addproduct
- 1.3 Admin - updateproduct
- 1.4 Admin - deleteproduct

Server

- Update 'server/models/ProductDAO.js' file

```
1  ...
2  const ProductDAO = {
3    ...,
4    async delete(_id) {
5      const result = await Models.Product.findByIdAndRemove(_id);
6      return result;
7    }
8  };
9  ...
```


Server

- Update 'server/api/admin.js' file

```
1  ...
2  // product
3  router.delete('/products/:id', JwtUtil.checkToken, async function (req, res) {
4      const _id = req.params.id;
5      const result = await ProductDAO.delete(_id);
6      res.json(result);
7  });
8  ...
```

- Postman: (DELETE) `http://localhost:3000/api/admin/products/<id>`
 - Headers: "x-access-token": <token>

Client-admin





React App

localhost:3001/admin/product

HOME CATEGORY PRODUCT ORDER CUSTOMER

Hello admin | Logout

PRODUCT LIST

ID	Name	Price	Creation date	Category	Image
6288bb8505f9bb7c9f45cc18	iPad Air	10	5/21/2022, 5:14:29 PM	iPad	
6288bd295ddf350614f2a916	iPad Pro	30	5/21/2022, 5:21:29 PM	iPad	
6288bc0940a1d1225e22805d	iPad Mini	20	5/21/2022, 5:16:41 PM	iPad	
6288bdda3afd8004d22ad823	iPhone 11 Pro Max	40	5/21/2022, 5:24:26 PM	iPhone	

1 | 2 | 3

PRODUCT DETAIL

ID

6288bb8505f9bb7c9f45

Name

iPad Air

Price

10

Image

Choose File

No file chosen

Category

iPad

ADD NEW

UPDATE

DELETE

