

MỤC TIÊU:

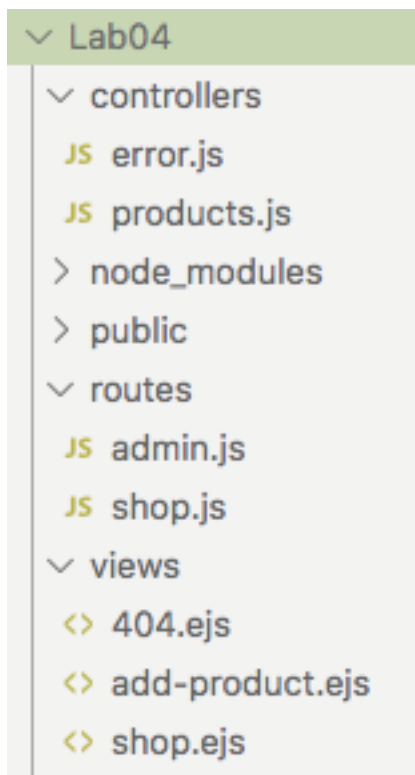
Kết thúc bài thực hành này bạn có khả năng

- ✓ Biết cách tổ chức code với mô hình MVC
- ✓ Xây dựng được model
- ✓ Xây dựng tốt controller
- ✓ Cài đặt sequelize

PHẦN I

Bài 1 (2 điểm)

Tổ chức website sách theo mô hình MVC



1. Tổ chức controller product

```
const products = [];  
exports.getAddProduct = (req, res, next) => {  
  res.render('add-product', {  
    pageTitle: 'Add Product',
```

```
    path: '/admin/add-product',
    activeAddProduct: true
  });
};

exports.postAddProduct = (req, res, next) => {
  products.push({ title: req.body.title });
  res.redirect('/');
};

exports.getProducts = (req, res, next) => {
  res.render('shop', {
    prods: products,
    pageTitle: 'Shop',
    path: '/',
    hasProducts: products.length > 0,
    activeShop: true,
  });
};
};
```

2. Route cho phần addmin

```
const express = require('express');
const productsController = require('../controllers/products');
const router = express.Router();
// /admin/add-product => GET
router.get('/add-product', productsController.getAddProduct);
// /admin/add-product => POST
router.post('/add-product', productsController.postAddProduct);
module.exports = router;
```

Phần shop sinh viên dựa vào phần admin trên để xây dựng route

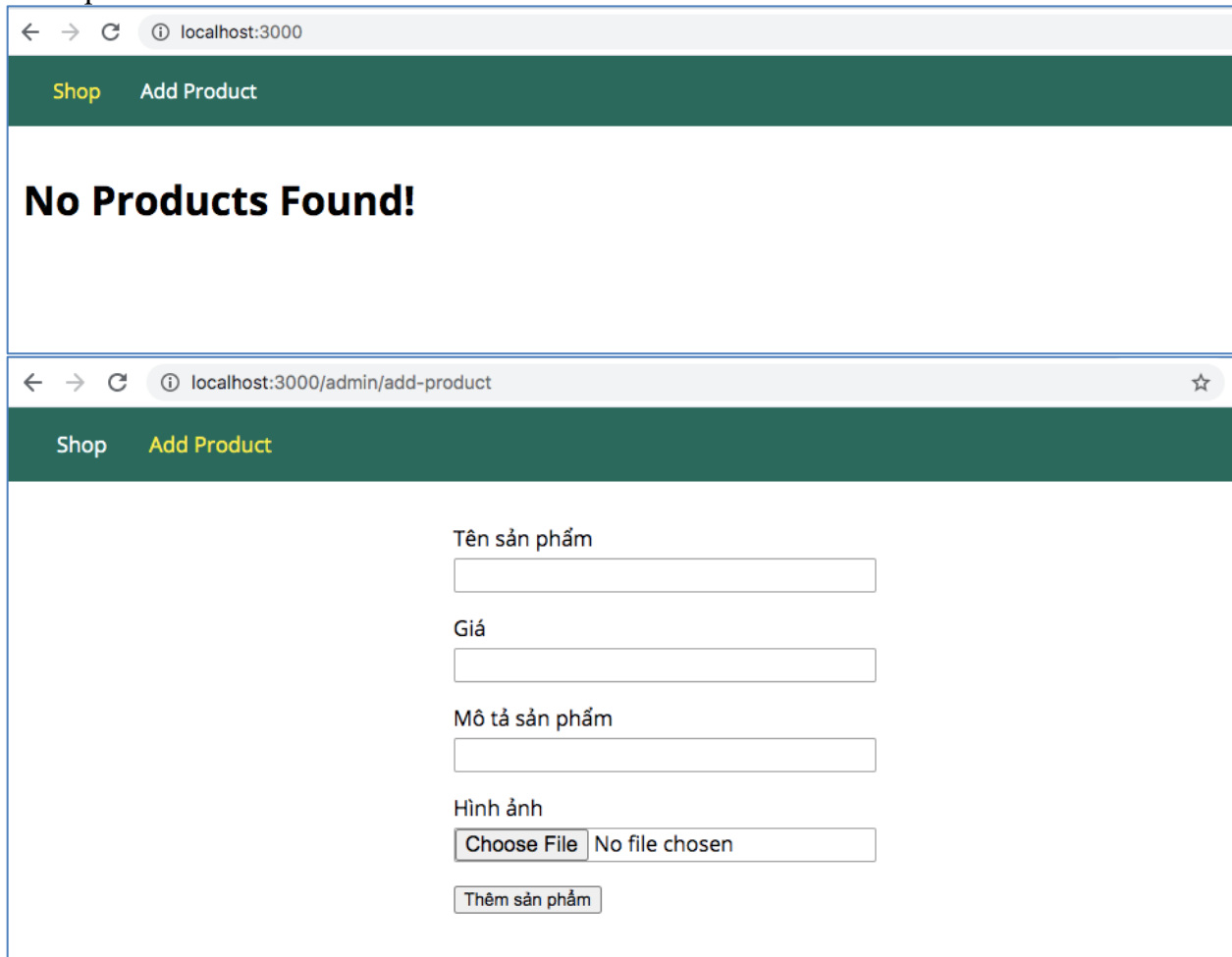
3. Phần server: sử dụng hệ thống route đã xây dựng bước trên

```
const adminRoutes = require('./routes/admin');
const shopRoutes = require('./routes/shop');
const errorController = require('../controllers/error');

app.use('/admin', adminRoutes);
```

```
app.use(shopRoutes);
```

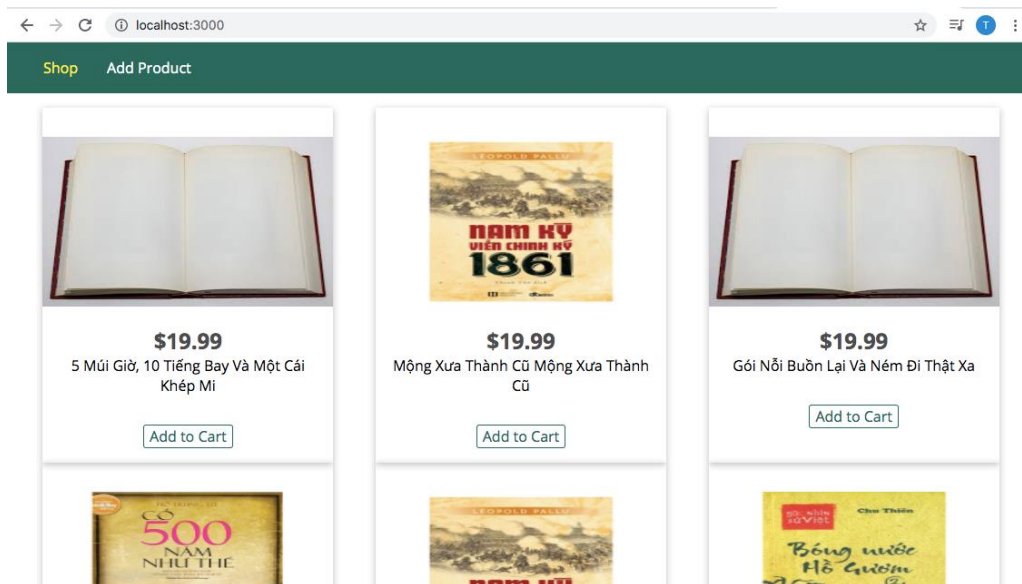
Kết quả:



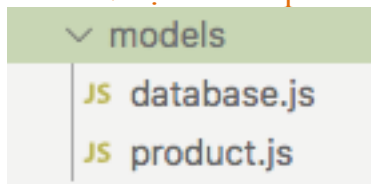
The screenshot displays two browser windows. The first window, at `localhost:3000`, shows a navigation bar with 'Shop' and 'Add Product' links, and a main content area with the text 'No Products Found!'. The second window, at `localhost:3000/admin/add-product`, shows the same navigation bar with 'Add Product' highlighted, and a form for adding a new product. The form includes input fields for 'Tên sản phẩm' (Product Name), 'Giá' (Price), and 'Mô tả sản phẩm' (Product Description), a file upload section for 'Hình ảnh' (Image) with a 'Choose File' button and 'No file chosen' text, and a 'Thêm sản phẩm' (Add Product) button at the bottom.

Bài 2 (3 điểm)

Xây dựng trang Shop



Bước 1: Tạo model product



Database:

```
var mysql = require('mysql');
var db = mysql.createConnection({
  host: 'localhost', // your host name
  user: 'root',      // database username
  password: '',      // database password
  database: 'book' //database Name
});
db.connect(function(err) {
  if (err) throw err;
  console.log('Database is connected successfully !');
});
module.exports = db;
})
```

Product:

```
var db=require('./database');
var products=[];
module.exports = class Product {
  constructor() {
  }
  //thêm một sản phẩm
  static saveProduct() {
```

```

}
//trả về tất cả sản phẩm
static fetchAll() {
  let sql = `SELECT * FROM products`;

  db.query(sql, function(err, data) {
    if (err) throw err;
    products=data;
  });
  return products;
}
}

```

Bước 2: Gọi model product tại controller product. Cập nhật phương thức getProducts như sau

```

const Product = require('../models/product');
exports.getProducts = (req, res, next) => {
  const products = Product.fetchAll();
  res.render('shop', {
    prods: products,
    pageTitle: 'Shop',
    path: '/',
    activeShop: true,
  });
};

```

Bước 3: view shop

```

<main align="center">
  <div class="grid">
    <% for (let product of prods) { %>
      <article class="card product-item">
        <header class="card__header">
          <h1 class="product__title"><%= product.nameProduct %></h1>
        </header>
        <div class="card__image">
          
        </div>
        <div class="card__content">
          <h2 class="product__price">$<%= product.priceProduct %></h2>
          <p class="product__description"><%=product.sortDescription %></p>
        </div>
        <div class="card__actions">
          <button class="btn">Add to Cart</button>
        </div>
      </article>
    }
  </div>
</main>

```

```
<% } %>
</div>
</main>
```

PHẦN II

Bài 3 (2 điểm)

Xây dựng trang thêm sản phẩm khi người dùng chuyển sang phần quản trị sản phẩm (localhost:3000/admin)

Tên sản phẩm

Giá

Mô tả sản phẩm

Hình ảnh

Choose File No file chosen

Thêm sản phẩm

Hướng dẫn:

Tại Model product thực hiện bổ sung hàm(phương thức)

```
//thêm một sản phẩm
static saveProduct(product) {
  db.query('insert into products SET ?',product, function(err, data) {
    if (err) throw err;
    return true;
  })
}
```

Tại controller product bổ sung hành động (hàm,phương thức) dạng post

```
exports.postAddProduct = (req, res, next) => {
  const file = req.file
  let title=req.body.productName;
  let price=req.body.price;
  let description=req.body.description;
  let nameImage=file.filename;
  product={
    nameProduct:title,
    priceProduct:price,
    sortDescription:description,
    images:nameImage,
  }
  Product.saveProduct(product);
  res.redirect('/');
};
```

Tại router của admin, bổ sung thêm route dạng post

```
router.post('/add-product',upload.single('productImage'), productsController.postAddProduct);
```

- Kết quả khi thêm một sản phẩm mới

Trang chủ Sản phẩm Thêm sản phẩm

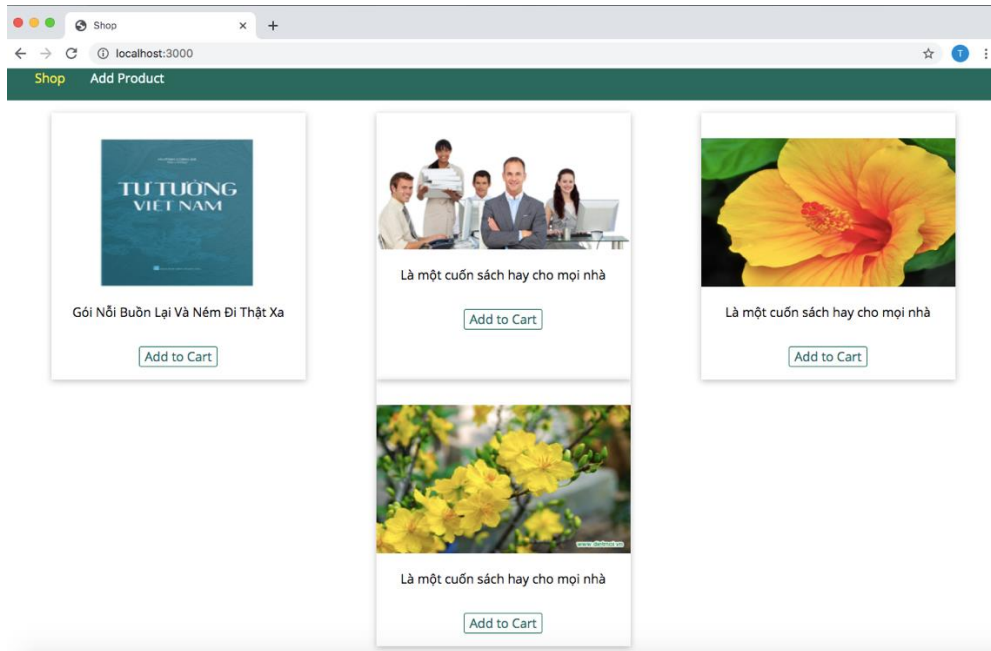
Tên sản phẩm
Sách hay cuộc sống

Giá
23000

Mô tả sản phẩm
Là một cuốn sách hay cho mọi nhà

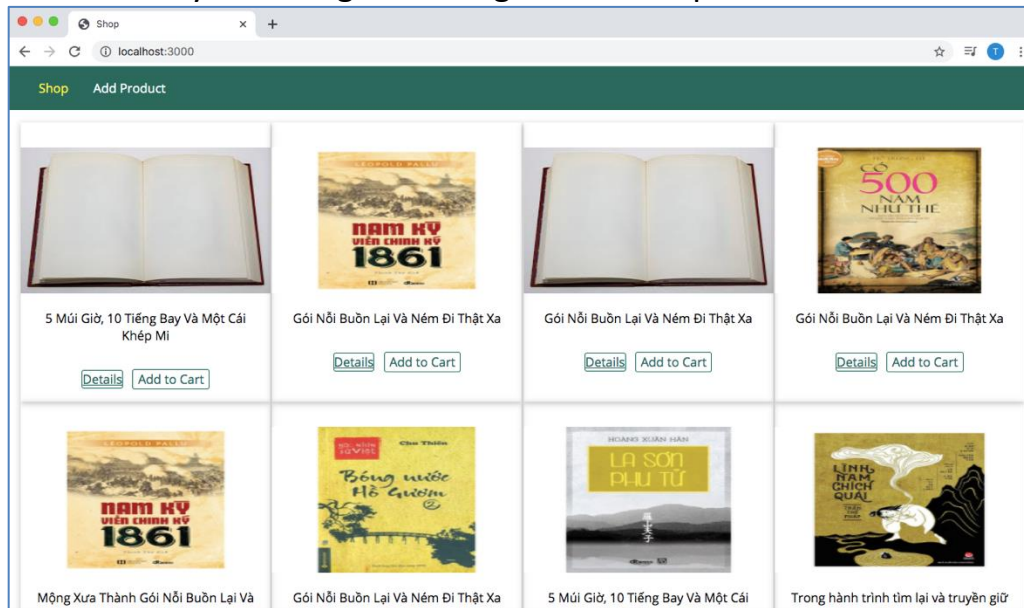
Hình ảnh
Choose File sach.png

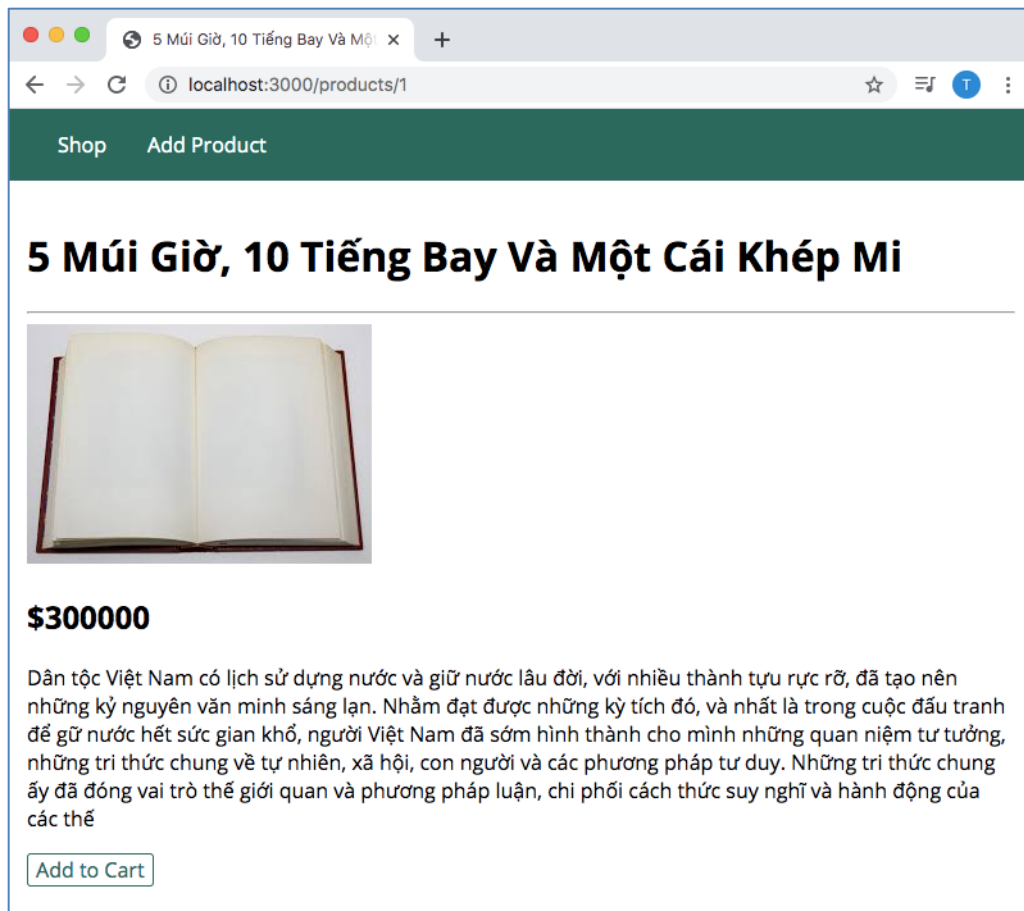
Thêm sản phẩm



Bài 4 (2 điểm)

Xây dựng trang xem chi tiết sản phẩm. Khi người dùng chọn sản phẩm cần xem, website chuyển hướng đến trang chi tiết sản phẩm





Hướng dẫn:

Bước 1: Cập nhật model product

```
//tìm sản phẩm theo id
static findById(id) {
  products=fetchAll();
  const product = products.find(p => p.id === id);
  return product;
}
```

Bước 2: Controller product cho trường hợp load chi tiết sản phẩm

```
//detail
exports.getProduct = (req, res, next) => {
  const prodId = req.params.productId;
  product=Product.findById(prodId);
  console.log(product);
  res.render('product-detail', {
    product: product,
    pageTitle: product.nameProduct,
  });
}
```

```
    path: '/products'  
  });  
};
```

Bước 3: Route cho trường hợp người dùng chuyển xem chi tiết

```
router.get('/products/:productId', productsController.getProduct);
```

Bước 4: Tạo liên kết trang sản phẩm – shop để xem chi tiết

```
<div class="card__actions">  
  <a href="/products/<%= product.idProduct %>" class="btn">Details</a>  
  <button class="btn">Add to Cart</button>  
</div>
```

Bài 5 (1 điểm)

Sinh viên làm tiếp phần quản trị sản phẩm cho 2 thao tác còn lại: Xóa sản phẩm và sửa sản phẩm

*** Yêu cầu nộp bài:

SV nén file (hoặc share thư mục google drive) bao gồm các yêu cầu đã thực hiện trên, nộp LMS đúng thời gian quy định của giảng viên. KHÔNG NỘP BÀI COI NHƯ KHÔNG CÓ ĐIỂM.

--- Hết ---