**Giải thích chi tiết đoạn mã**

Đoạn mã này định nghĩa một **controller** để xử lý các yêu cầu liên quan đến trang chủ và các trang con của ứng dụng. Nó sử dụng mô hình **MVC (Model-View-Controller)** trong Node.js với Sequelize ORM.

**1. Khởi tạo Controller**

const controller = {};

* **controller** là một đối tượng chứa các phương thức để xử lý logic liên quan đến các yêu cầu HTTP.

**2. Phương thức showHomepage**

**Chức năng**

* Lấy dữ liệu từ cơ sở dữ liệu, bao gồm:
  + **Sản phẩm nổi bật** (Featured Products).
  + **Sản phẩm mới** (Recent Products).
  + **Danh mục sản phẩm** (Categories).
  + **Thương hiệu** (Brands).
* Gửi dữ liệu này đến view để hiển thị.

**Phân tích từng bước**

const featuredProducts = await models.Product.findAll({

attributes: ['id', 'name', 'imagePath', 'stars','price','oldPrice'],

order:[['stars','DESC']],

limit: 10

});

res.locals.featuredProducts = featuredProducts;

* **models.Product.findAll**:
  + Truy vấn bảng Product để lấy danh sách sản phẩm nổi bật.
  + **attributes**: Chỉ chọn các cột cần thiết như id, name, imagePath, stars, price, và oldPrice.
  + **order**: Sắp xếp theo số sao (**stars**) giảm dần (**DESC**).
  + **limit**: Giới hạn số lượng sản phẩm là 10.
* **res.locals.featuredProducts**:
  + Lưu trữ dữ liệu để truyền sang view.

const recentProducts = await models.Product.findAll({

attributes: ['id', 'name', 'imagePath', 'stars','price','oldPrice','createdAt'],

order:[['createdAt','DESC']],

limit: 10

});

res.locals.recentProducts = recentProducts;

* Tương tự như trên, nhưng truy vấn sản phẩm mới nhất:
  + Sắp xếp theo **createdAt** (thời gian tạo) giảm dần (**DESC**).
  + Lưu trữ kết quả vào **res.locals.recentProducts**.

const categories = await models.Category.findAll();

const secondArrayy = categories.splice(2,2);

const thirdArray = categories.splice(1,1);

res.locals.categaryArray = [categories,secondArrayy,thirdArray];

* **models.Category.findAll**:
  + Lấy toàn bộ danh sách danh mục từ bảng Category.
* **splice**:
  + Chia danh sách danh mục thành ba nhóm:
    - **categories**: Nhóm đầu tiên sau khi loại bỏ các danh mục khác.
    - **secondArrayy**: Nhóm thứ hai, bao gồm 2 danh mục từ vị trí thứ 2.
    - **thirdArray**: Nhóm thứ ba, bao gồm 1 danh mục từ vị trí thứ 1.
* **res.locals.categaryArray**:
  + Lưu ba nhóm danh mục vào một mảng để truyền sang view.

const Brand = models.Brand;

const brands = await Brand.findAll();

res.render('index',{brands:brands});

* Lấy danh sách thương hiệu từ bảng Brand.
* **res.render**:
  + Kết hợp dữ liệu đã chuẩn bị với view **index**.
  + Truyền danh sách thương hiệu vào view qua đối số **{brands: brands}**.

**3. Phương thức showPage**

**Chức năng**

* Xử lý yêu cầu hiển thị các trang con của ứng dụng (ví dụ: cart, checkout, product...).

**Phân tích**

const pages =['cart','checkout','product','login','wishlist','contact','product-detail','product-list','my-account'];

if(pages.includes(req.params.page))

return res.render(req.params.page);

netx();

* **pages**:
  + Danh sách các trang con hợp lệ.
* **req.params.page**:
  + Lấy tên trang từ tham số URL.
* **includes**:
  + Kiểm tra xem tên trang có nằm trong danh sách hợp lệ hay không.
* **res.render(req.params.page)**:
  + Nếu hợp lệ, hiển thị trang tương ứng.
* **next()**:
  + Nếu không hợp lệ, chuyển xử lý sang middleware tiếp theo.

**4. Xuất Controller**

module.exports = controller;

* Xuất đối tượng **controller** để sử dụng trong các route.

**Tóm tắt**

* **showHomepage**: Chuẩn bị dữ liệu cho trang chủ, bao gồm sản phẩm nổi bật, sản phẩm mới, danh mục, và thương hiệu.
* **showPage**: Hiển thị các trang con nếu hợp lệ, hoặc chuyển tiếp nếu không.
* Kết hợp với Sequelize để truy vấn cơ sở dữ liệu và sử dụng **res.locals** để truyền dữ liệu vào view.