**1. 'use strict';**

* Kích hoạt chế độ nghiêm ngặt của JavaScript để tránh các lỗi thông thường, cải thiện hiệu suất và tránh các hành vi không an toàn.

**2. Import các module và khai báo biến**

javascript

Copy code

const express = require('express');

const app = express();

const port = process.env.PORT || 5000;

const expressHandlebars = require('express-handlebars');

const {createStarList} = require('./controller/handlebarsHelper');

const {createPagination} = require('express-handlebars-paginate');

const session = require('express-session');

* **express**: Framework để xây dựng ứng dụng web.
* **app**: Ứng dụng Express.
* **port**: Cổng mà ứng dụng sẽ chạy, có thể được cấu hình qua biến môi trường hoặc mặc định là 5000.
* **expressHandlebars**: Dùng để cấu hình và sử dụng **Handlebars** làm engine render giao diện.
* **createStarList**, **createPagination**: Các hàm helper để hỗ trợ render giao diện.
* **session**: Quản lý session cho người dùng.

**3. Cấu hình tệp tĩnh**

javascript

Copy code

app.use(express.static(\_\_dirname + '/public'));

* Cấu hình thư mục chứa các tệp tĩnh (CSS, JS, hình ảnh) là thư mục /public.

**4. Cấu hình Handlebars**

javascript

Copy code

app.engine('hbs', expressHandlebars.engine({

layoutsDir: \_\_dirname + '/views/layouts',

partialsDir: \_\_dirname + '/views/partials',

extname: 'hbs',

defaultLayout: 'layout',

runtimeOptions: {

allowProtoPropertiesByDefault: true,

},

helpers: {

createStarList,

createPagination

}

}));

app.set('view engine', 'hbs');

app.set('views', \_\_dirname + '/views');

* **app.engine('hbs', ...)**: Đăng ký **Handlebars** làm engine render với phần mở rộng .hbs.
* **layoutsDir**: Thư mục chứa các file layout.
* **partialsDir**: Thư mục chứa các phần giao diện dùng chung (partials).
* **extname**: Phần mở rộng của các file template là .hbs.
* **defaultLayout**: Layout mặc định là layout.hbs.
* **helpers**: Các hàm hỗ trợ render giao diện như createStarList và createPagination.
* **runtimeOptions**: Cho phép các thuộc tính và phương thức của prototype được sử dụng trong Handlebars.

**5. Cấu hình để đọc dữ liệu từ POST request**

javascript

Copy code

app.use(express.json());

app.use(express.urlencoded({extended: false}));

* **express.json()**: Xử lý các request có body dạng JSON.
* **express.urlencoded({extended: false})**: Xử lý request có body dạng URL-encoded.

**6. Cấu hình session**

javascript

Copy code

app.use(session({

secret:'S3cret',

resave:false,

saveUninitialized:false,

cookie: {

httpOnly:true,

maxAge: 20\*60\*1000 // 20 phút

}

}));

* **secret**: Chuỗi bí mật dùng để mã hóa session.
* **resave**: Không lưu lại session nếu không thay đổi.
* **saveUninitialized**: Không lưu session mới chưa được khởi tạo.
* **cookie**: Cấu hình cookie cho session, với thời gian sống là 20 phút.

**7. Middleware quản lý giỏ hàng**

javascript

Copy code

app.use((req, res, next) => {

let Cart = require('./controller/cart');

req.session.cart = new Cart(req.session.cart ? req.session.cart : {});

res.locals.quantity = req.session.cart.quantity;

next();

});

* **req.session.cart**: Tạo một giỏ hàng (Cart) mới nếu chưa tồn tại, hoặc lấy từ session hiện tại.
* **res.locals.quantity**: Gán số lượng sản phẩm trong giỏ hàng vào biến cục bộ để sử dụng trong giao diện.

**8. Định tuyến (Routing)**

javascript

Copy code

app.use('/', require('./routes/indexRouter'));

app.use('/products', require('./routes/productsRouter'));

* **indexRouter**: Xử lý các route của trang chủ.
* **productsRouter**: Xử lý các route liên quan đến sản phẩm.

**9. Xử lý lỗi**

javascript

Copy code

app.use((req, res, next) => {

res.status(404).render('error', {message: 'File not found'});

});

app.use((error, req, res, next) => {

console.log(error);

res.status(500).render('error', {message: 'Internal server error'});

});

* **404 Error**: Khi không tìm thấy route, render giao diện lỗi với thông báo "File not found".
* **500 Error**: Khi xảy ra lỗi server, ghi log lỗi và render giao diện lỗi.

**10. Khởi động server**

javascript

Copy code

app.listen(port, () => {

console.log(`Server is running at http://localhost:${port}`);

});

* **app.listen(port)**: Khởi chạy server và lắng nghe trên cổng được chỉ định.

**Tóm tắt:**  
Đoạn mã này xây dựng một ứng dụng web với cấu trúc rõ ràng, sử dụng các thành phần:

* **Express** để định tuyến và xử lý request.
* **Handlebars** để render giao diện.
* **Session** để quản lý trạng thái người dùng (ví dụ: giỏ hàng).
* Có các chức năng xử lý lỗi và cấu hình đầy đủ cho việc phát triển ứng dụng.