

Design concept

1. Coupling

1.1. Content coupling

Related module	Description	Improvement direction
OrderController và các Repository	OrderController phụ thuộc trực tiếp vào nhiều repository như IProvinceRepository, IDistrictRepository, IWardRepository, IMediaRepository, và IOrderRepository. Điều này làm tăng coupling	Tách phần repository này vào một Service Layer. Controller chỉ nên gọi các service thay vì làm việc trực tiếp với repository
OrderController và IVnPayService	Controller trực tiếp tương tác với IVnPayService để xử lý thanh toán, gây ra sự phụ thuộc mạnh vào một service cụ thể	Đưa logic thanh toán vào 1 lớp PayOrderController. Controller chỉ nhận kết quả trả về từ lớp này
CartController và IMediaRepository	CartController phụ thuộc trực tiếp vào repository IMediaRepository trong nhiều phương thức như ProcessOrderDirectly, AddToCart, và CheckStock	Tách logic xử lý sang lớp Service, controller chỉ gọi Service thay vì làm việc trực tiếp với repository
HttpContext.Session	Nhiều phương thức (AddToCart, GetCartFromSession, SaveCartToSession, ProcessOrderDirectly, v.v.) phụ thuộc trực tiếp vào HttpContext.Session	Tạo một quản lý session để trừu tượng hóa việc thao tác dữ liệu trong session
HomeController và IMediaRepository	HomeController phụ thuộc trực tiếp vào IMediaRepository trong các phương thức như Index, SearchResultView, MediaDetailsView	Tách logic truy vấn dữ liệu sang lớp Service, controller sẽ gọi Service thay vì làm việc trực tiếp với repository

1.2. Control coupling

Related module	Description	Improvement direction
OrderController.SaveOrderData và Client	Phương thức SaveOrderData yêu cầu rất nhiều tham số từ phía client (province, district, ward, shippingMethod, ...) để hoạt động đúng cách	Giảm bớt số lượng tham số bằng cách chuyển sang sử dụng 1 lớp OrderRequestModel để đóng gói tất cả dữ liệu đầu vào
SearchResultView và tham số	Phương thức SearchResultView nhận nhiều tham số (searchTerm, category, sortBy) và truyền chúng qua ViewBag để sử dụng trong View.	Sử dụng một lớp để đóng gói các tham số và truyền chúng vào View một cách rõ ràng

2. Cohesion

2.1. Coincidental cohesion

Related module	Description	Improvement direction
OrderController	OrderController xử lý quá nhiều logic không liên quan, từ quản lý phiên (session) đến xử lý thanh toán.	Tách riêng các logic này ra các lớp riêng biệt, SessionManager, PayOrderController
CartController	CartController thực hiện nhiều nhiệm vụ khác nhau: quản lý giỏ hàng, kiểm tra tồn kho, xử lý đơn hàng.	Tách các chức năng này thành các controller hoặc lớp riêng biệt

HomeController	HomeController đảm nhận nhiều nhiệm vụ khác nhau: xử lý trang chủ, tìm kiếm, chi tiết phương tiện, quyền riêng tư	Chia nhỏ controller theo trách nhiệm

2.2. Procedural cohesion

Related module	Description	Improvement direction
CalculateShippingFee	Phương thức này chỉ tập trung vào tính toán phí vận chuyển dựa trên dữ liệu nhưng không kết hợp chặt chẽ với các phương thức khác.	Chuyển phương thức này sang một lớp khác như ShippingService để tăng gắn kết với các nghiệp vụ liên quan
UpdateCart	Phương thức UpdateCart chỉ tính tổng tiền và trả về mà không liên kết chặt chẽ với các phương thức khác trong module.	Đưa logic tính tổng tiền sang một lớp riêng để tăng sự gắn kết với các nghiệp vụ liên quan
MediaDetailsView	Phương thức sử dụng cấu trúc lệnh if-else để xử lý từng loại phương tiện (DVD, CD, Book)	Tách logic xử lý riêng cho từng loại vào các phương thức hoặc lớp riêng biệt
SortMedias	Phương thức xử lý logic sắp xếp nhưng không kết nối chặt chẽ với các nghiệp vụ khác trong HomeController.	Chuyển sang lớp khác như SortMedia để tăng sự gắn kết và tái sử dụng