**📑 ASP.NET Core 8 MVC Project Checklist (Database mới có ProductCategory)**

**Phase 1 — Identity & Login**

**B1. Khởi tạo MVC cơ bản**

Trong Program.cs: AddControllersWithViews(), pipeline UseHttpsRedirection, UseStaticFiles, UseRouting, MapDefaultControllerRoute().

Tạo HomeController(Index) + Views/Home/Index.cshtml hiển thị “OK”.

**B2. Chuỗi kết nối SQL Server**

Thêm "ConnectionStrings:DefaultConnection" vào appsettings.json:

"Server=.;Database=ShopDbMvc;Trusted\_Connection=True;TrustServerCertificate=True;MultipleActiveResultSets=true"

**B3. DbContext & gói EF Core**

Tạo ApplicationDbContext kế thừa IdentityDbContext<ApplicationUser>.

Đăng ký trong Program.cs với UseSqlServer.

Cài gói nuget: Microsoft.EntityFrameworkCore.SqlServer, Microsoft.EntityFrameworkCore.Tools, Microsoft.AspNetCore.Identity.EntityFrameworkCore.

**B4. FULL Identity + ApplicationUser**

Tạo ApplicationUser : IdentityUser (thêm CurrentShopId long?, CurrentBranchId long?).

Program.cs: AddDefaultIdentity<ApplicationUser>().AddRoles<IdentityRole>().AddEntityFrameworkStores<ApplicationDbContext>().

**B5. AccountController + Views**

Tạo AccountController (Login/Register/Logout) + Views.

Dùng UserManager, SignInManager. Thêm [ValidateAntiForgeryToken] cho POST.

**B6. Migration “IdentityOnly”**

dotnet ef migrations add IdentityOnly

dotnet ef database update

**Phase 2 — Domain Database & Seed**

**B7. Entity & DbContext**

Sinh các entity C# và DbSet theo database mới:

- Shop, Branch

- ProductCategory (có ParentCategory, Shop, Code unique)

- Brand, Product (liên kết Category, Brand, Shop; có RowVersion, IsDeleted, ImagePath)

- ProductVariant (Sku unique per Product; TrackSerial; RowVersion)

- Inventory, InventorySerial

- Employee (liên kết AspNetUsers)

- Customer (Code unique per Shop, IsArchived, RowVersion)

- Coupon, CouponProduct, CouponRedemption

- OrderStatus, Order, OrderLine, OrderCoupon, OrderLineSerial

- PaymentMethod, Payment

- SubscriptionPlan, ShopSubscription, BillingTransaction

- ErrorLog

Thêm Fluent API đầy đủ.

**B8. Migration “DomainInitial” + Update DB**

dotnet ef migrations add DomainInitial

dotnet ef database update

**B9. Seed Data**

Seed Roles: SuperAdmin, ShopOwner, Cashier, Sale

Seed OrderStatus, PaymentMethod

Seed Shop demo (SHOP01), Branch demo (BR01)

Seed ProductCategory demo: Điện thoại, Laptop, Phụ kiện

Seed Brand demo: Apple, Samsung

Seed Product demo (iPhone 15) + Variant demo (128GB)

Seed Inventory demo (10 sản phẩm)

Seed Customer demo (Nguyễn Văn A)

Seed Coupon demo (SALE10, 10%, 30 ngày)

**Phase 3 — CRUD Controllers & Views**

**B10. ProductCategory CRUD**

Tạo ProductCategoriesController + Views:

- Index: filter ShopId, search Code/Name; paging

- Create/Edit: validate unique(ShopId, Code); chọn ParentCategory

- Delete: chặn nếu có Product tham chiếu

**B11. Brand CRUD**

Tạo BrandsController + Views:

- Index: filter ShopId, search Name

- Create/Edit: Name unique per Shop

- Delete: chặn nếu có Product tham chiếu

**B12. Product CRUD**

Tạo ProductsController + Views:

- Index: filter ShopId, CategoryId, BrandId, CanSell, keyword; paging; thumbnail

- Create/Edit: chọn Category + Brand; validate Code unique per Shop; upload ảnh .jpg/.png <=1MB

- Delete: soft-delete (IsDeleted=1); có action Restore

**B13. ProductVariant CRUD (AJAX)**

Tạo ProductVariantsController (API) + Partial \_VariantsTable.cshtml + JS fetch:

- CRUD Variant inline trong Product Details

- Unique Sku per Product; chặn delete nếu TrackSerial=1 và có InventorySerial

**B14. Customer CRUD**

Tạo CustomersController + Views:

- Index: search Phone/Email/Code/Name; filter có/không TaxCode

- Create/Edit: validate Phone/Email; auto-gen Code "CUS-yyMMdd-####"

- Delete: nếu có Order -> IsArchived=1

**B15. Employee CRUD**

Tạo EmployeesController + Views:

- Index: filter BranchId, IsActive, keyword

- Create/Edit: chọn User (AspNetUsers typeahead); chọn Branch; unique(ShopId, UserId)

- Delete: nếu có Order.SalesUserId thì IsActive=false

**B16. Coupon CRUD**

Tạo CouponsController + Views:

- Index: filter thời gian, IsActive, Type, keyword

- Create/Edit: validate StartAt<EndAt, Value>0; MultiSelect ProductVariant (AJAX)

- Delete: nếu đã redeem thì IsActive=false

**B17. Inventory**

Tạo InventoryController:

- Index: join Inventory + ProductVariant + Product + Branch; filter Branch, Category, keyword; sort; paging

- Serials(variantId): liệt kê serial theo Status

**B18. Orders & Payments**

Tạo OrdersController + Views:

- Index: filter ngày/trạng thái/khách

- Create: form thêm dòng hàng động (JS); tính SubTotal/Discount/Tax/Total; nhập Coupon

- Details: hiển thị serial đã bán

- Ship: nhập serial per line nếu TrackSerial

- Payments: nhận thanh toán, hiển thị lịch sử Payment

**Phase 4 — Services**

**B19. InventoryService / CouponService / OrderWorkflowService**

Tạo service Scoped:

InventoryService

- AllocateAsync(orderId), ReleaseAsync(orderId), ShipAsync(orderId, serialsByLine)

CouponService

- Validate (thời gian, IsActive, MinOrderAmount, MaxRedemptions, PerCustomerLimit, ProductVariant áp dụng)

- Tính DiscountAmount, ghi OrderCoupon + CouponRedemption

OrderWorkflowService

- ConfirmAsync -> Allocate

- ShipAsync -> Ship + update Status

- CancelAsync -> Release

**Phase 5 — Reports & ErrorLog**

**B20. Reports**

Map rpt.RevenueDaily, rpt.PaymentHistory, rpt.PersonalSales thành entity keyless.

Tạo ReportsController:

- Dashboard: KPI hôm nay, biểu đồ 30 ngày, PaymentHistory gần đây

- MySales: doanh số cá nhân

**B21. ErrorLog**

Tạo ErrorLoggingMiddleware: catch lỗi, ghi ErrorLog(Level, Source, Message, Exception, UserId, ShopId).

Tạo LogsController + Views: filter Level/Time/User/Shop, paging, export CSV.

**Phase 6 — Phân quyền**

**B22. Feature/RoleFeature**

Tạo bảng Feature, RoleFeature, seed toàn bộ feature codes.

**B23. Seed Role-Feature Mapping**

Tạo SeedIdentityFeature:

- 4 Roles (SuperAdmin, ShopOwner, Cashier, Sale)

- Seed toàn bộ Feature codes

- Map Role->Feature theo thiết kế

- Gán admin@local vào SuperAdmin

**B24. Claims Transformation**

Triển khai IClaimsTransformation:

- Lấy Roles -> query RoleFeature(IsEnabled=1) -> add claim("feature", code)

**B25. Authorization Policies**

AddAuthorization trong Program.cs: tạo Policy cho từng Feature code.

Áp dụng [Authorize(Policy="...")] vào Controllers/Actions.

**B26. UI Helper**

Tạo TagHelper/HtmlHelper PolicyButton: nếu Authorized(policy) thì render nút/link, ngược lại ẩn.

Dùng trong \_Layout và các View Index.