

1) Tổng quan & phạm vi

- **Hình thức:** Làm theo **nhóm 3–5 SV**.
- **Sản phẩm:** 01 Web App **Client–Server** (SPA frontend + API backend).
- **Công nghệ gợi ý:**
 - **Frontend:** React, Vue (SPA, router, state management).
 - **Backend:** Express (Node.js) hoặc ASP.NET Core.
 - **ORM:** Prisma / TypeORM / Sequelize (Node) **hoặc** EF Core (ASP.NET).
 - **Database:** MongoDB, MySQL, PostgreSQL, **hoặc** Firebase Firestore/Auth).
- **Mẫu đề tài gợi ý (chọn 1 hoặc tự đề xuất với Giảng viên hướng dẫn):**
 - E-Commerce mini
 - Event/Ticket Booking
 - Learning/LMS
 - Task/Project Management
 - Blog/CMS nâng cao
 - Library/Inventory
 - Clinic/Appointment
 - Hotel/Room Booking...

2) Yêu cầu chức năng (Functional Requirements)

Tối thiểu đáp ứng các mục sau (có thể mở rộng):

1. Xác thực người dùng và Phân quyền

- Đăng ký/đăng nhập, **hash password** (bcrypt/Argon2) hoặc Firebase Auth.
- **Role-based access** tối thiểu: admin & user.
- **JWT** hoặc Session-based.

2. Quản lý dữ liệu cốt lõi

- Ít nhất **3 thực thể** có quan hệ (Ví dụ: User–Order–OrderItem hoặc Blog–Post–Tag).
- Đầy đủ **CRUD; search, sort, pagination** (server-side).
- **Upload file** (ảnh đại diện/sản phẩm/tài liệu) và hiển thị.

3. Báo cáo/Thông kê

- 01 trang dashboard (biểu đồ/tổng hợp), ví dụ: doanh thu theo tháng, số đơn theo trạng thái, top N item...

3) Yêu cầu phi chức năng (Non-Functional Requirements)

- **API Design:** RESTful (hoặc GraphQL). Sử dụng **OpenAPI/Swagger UI** (hoặc tương đương) để duyệt & thử API.
- **Bảo mật cơ bản:**
 - Xác thực và chuẩn hóa dữ liệu từ người dùng; chống **XSS/SQLi**; cấu hình **CORS** hợp lý.
- **Dữ liệu & Seed:** Script tạo schema/migration và **seed data** tối thiểu.

4) Tài liệu & Mã nguồn (Deliverables)

Bắt buộc trong GitHub

- **Một repo public** (hoặc 2 repo FE/BE).
- **README.md** ở root, gồm:
 1. Mô tả ngắn, tính năng chính, ảnh chụp màn hình.
 2. Kiến trúc (sơ đồ tổng thể, tech stack, lý do chọn).
 3. Hướng dẫn chạy nhanh (local & Docker), biến môi trường, seeding.
 4. Tài khoản demo (user/admin), link Swagger, link web deploy (nếu có).
 5. Cấu trúc thư mục, conventions (coding style, commit, branch).
 6. Kịch bản demo (use cases chính, đường dẫn UI & API).
- **API docs:** OpenAPI/Swagger JSON/YAML + đường dẫn UI.
- **DB schema:** ERD (ảnh hoặc nguồn từ tool) + migration scripts.

Bài tiểu luận (Docx + PDF)

- Trang bìa (tên đề tài, thành viên, lớp, GV), Tóm tắt (Abstract).
 1. Giới thiệu & mục tiêu;
 2. Phân tích yêu cầu;
 3. Thiết kế hệ thống (kiến trúc, ERD, luồng auth, API chính);
 4. Cài đặt & công nghệ;
 5. Đánh giá (chức năng, hiệu năng, bảo mật, UX, hạn chế);
 6. Kết luận & hướng phát triển;

7. Tài liệu tham khảo;