**YÊU CẦU ĐỒ ÁN MÔN HỌC:   
CÔNG NGHỆ PHẦN MỀM NÂNG CAO**

**1. Mục tiêu đồ án**

Sinh viên thực hiện một **dự án phần mềm nhóm** theo quy trình **Agile/Scrum**, sử dụng các công cụ và công nghệ hiện đại trong quản lý, phát triển, triển khai phần mềm. Đồ án không chỉ đánh giá năng lực lập trình mà còn kiểm tra khả năng **làm việc nhóm, quản lý dự án phần mềm, thiết kế kiến trúc và triển khai hệ thống thực tế**.

**2. ‍Thành viên nhóm**

* Mỗi nhóm gồm **2 đến 3 học viên**.
* Có phân công vai trò rõ ràng: **Scrum Master**, **Developer**, **Product Owner** (có thể kiêm nhiệm nếu ít người).

**3. Yêu cầu kỹ thuật & quy trình**

**🌀 Agile/Scrum**

* Áp dụng mô hình **Scrum**:
  + Lập kế hoạch sprint 1–2 tuần/lần (ít nhất 3 sprint).
  + Có họp **Daily Scrum** và ghi nhận trên Jira.
  + Có Sprint Review và Retrospective sau mỗi vòng.

**📋 Jira – Quản lý backlog**

* Sử dụng **Jira Software (Cloud)** để:
  + Tạo **Product Backlog**: liệt kê user stories và tasks.
  + Tạo các sprint và quản lý tiến độ.
  + Thực hiện phân công công việc.
  + Đính kèm ảnh chụp màn hình (screenshot) trong báo cáo để minh chứng.

**🔀 Git & GitHub**

* Mỗi nhóm tạo 1 repository **công khai hoặc private (có mời giảng viên)**.
* Tuân thủ:
  + Mỗi tính năng trên một branch riêng.
  + Pull request (PR) để merge.
  + Ghi lại lịch sử commit, thể hiện sự đóng góp từng thành viên.
  + Thực hiện review mã nguồn trong các PR.

**🏗️ Thiết kế kiến trúc phần mềm**

* Áp dụng kiến trúc:
  + **Phân tầng (multi-layer)**: controller – service – repository – model.
  + Hoặc **microservices** (nếu nhóm đủ năng lực).
* Giao tiếp qua **RESTful API** (JSON).

**🐳 Triển khai bằng Docker**

* Đóng gói ứng dụng bằng Docker.
* Chạy được ứng dụng qua Docker Compose hoặc Docker CLI.
* Tùy chọn nâng cao: sử dụng **GitHub Actions** để CI/CD và **tự động deploy lên server** (có thể dùng Heroku, Render, Railway, hoặc VPS).

**4. Chủ đề gợi ý**

Sinh viên có thể chọn một trong các chủ đề sau hoặc đề xuất chủ đề riêng có sự phê duyệt:

* Hệ thống **quản lý sinh viên làm việc nhóm qua GitHub** (theo dõi commit, pull request, chấm điểm).
* **Ứng dụng RAG (Retrieval-Augmented Generation)** đơn giản với giao diện web.
* **Web app quản lý mã nguồn** (lưu trữ, xem lịch sử, tìm kiếm).
* Hệ thống **theo dõi tiến độ học tập**, tích hợp GitHub hoặc Google Docs API.
* **Hệ thống quản lý backlog** mô phỏng Jira.
* **Ứng dụng học trực tuyến mini**, phân quyền giảng viên – sinh viên.

**5. Cấu trúc báo cáo đồ án (theo chuẩn 5 chương)**

| **Chương** | **Nội dung** |
| --- | --- |
| **Chương 1: Giới thiệu** | Lý do chọn đề tài, mục tiêu, phạm vi, phương pháp và công cụ sử dụng |
| **Chương 2: Cơ sở lý thuyết & Công nghệ** | Giới thiệu về Agile/Scrum, GitHub, Docker, REST, Jira, CI/CD… |
| **Chương 3: Phân tích & Thiết kế hệ thống** | - Use case, sơ đồ hoạt động, ERD (nếu có DB) - Thiết kế kiến trúc phân tầng/microservice - Lưu đồ, sơ đồ class hoặc sequence diagram |
| **Chương 4: Thực hiện & Triển khai** | - Quản lý project bằng Jira (screenshot backlog, sprint) - Quản lý code bằng GitHub (PR, commit log) - Dockerfile và script triển khai - Kết quả chạy thực tế (ảnh minh họa) |
| **Chương 5: Đánh giá & Kết luận** | - Những gì đã làm được - Hạn chế - Hướng phát triển thêm - Phản hồi từ retrospective (nếu có) |

**6. Yêu cầu nộp**

* **Báo cáo PDF** theo mẫu 5 chương như trên.
* **Link GitHub repository** của nhóm.
* **Ảnh minh họa Jira**, **log commit**, **ảnh triển khai Docker**.
* **Video demo 5–10 phút** trình bày ứng dụng + CI/CD (nếu có).
* (Tùy chọn nâng cao): Nộp link web đã deploy.

**7. Thời hạn thực hiện**

* Thời gian: **4–6 tuần**.
* Báo cáo tiến độ giữa kỳ (có review Sprint).
* Nộp đồ án + thuyết trình cuối kỳ.