**ĐỒ ÁN CHUYÊN NGÀNH CÔNG NGHỆ THÔNG TIN**

**Kiến trúc tổng thể hệ thống**

Hệ thống được thiết kế theo mô hình **microservice hiện đại**, gồm các thành phần chính:

1. **Frontend (ReactJS)**

* Giao diện người dùng: dashboard nhóm, quản lý tài liệu, phân công công việc, lịch sử hoạt động
* Giao tiếp với backend qua REST API và WebSocket (nếu cần realtime)

2. **Backend (NodeJS + Express)**

* Xử lý logic nghiệp vụ: quản lý nhóm, phân quyền, lưu hoạt động
* Giao tiếp với MCP Server để truy xuất tài liệu và ngữ cảnh
* Gửi yêu cầu đến LLM để xử lý ngôn ngữ tự nhiên

3. **MCP Server (Model Context Protocol)**

* Giao thức trung gian giúp truy xuất tài liệu từ nhiều nguồn: Google Drive, OneDrive, hệ thống nội bộ
* Chuẩn hóa dữ liệu đầu vào cho LLM
* Có thể triển khai như một service riêng hoặc tích hợp vào backend

4. **LLM Service (GPT-4, Claude, hoặc API khác)**

* Nhận ngữ cảnh từ MCP và truy vấn từ người dùng
* Trả về câu trả lời, tóm tắt, hoặc phân tích tiến độ nhóm

5. **Database (MongoDB hoặc Firebase)**

* Lưu thông tin người dùng, nhóm, tài liệu, hoạt động, câu hỏi và câu trả lời
* Lưu lịch sử truy vấn để AI có thể “ghi nhớ” lâu dài



🧩 **Mô tả chi tiết các microservice**

1. **Auth Service** – Xác thực & Phân quyền

**Chức năng chính:**

* Đăng ký, đăng nhập, đăng xuất
* Xác thực người dùng bằng JWT hoặc OAuth2
* Phân quyền theo vai trò (thành viên, nhóm trưởng, quản trị hệ thống)
* Quản lý phiên đăng nhập (session management)
* Mã hóa mật khẩu (bcrypt/argon2)

**Dữ liệu xử lý:**

* Thông tin người dùng (userId, email, mật khẩu mã hóa, vai trò)
* Token truy cập và token làm mới (access/refresh token)

**Tương tác:**

* Cung cấp token cho các service khác để xác thực
* Kiểm tra quyền truy cập trước khi cho phép thao tác tài nguyên

2. **Group Service** – Quản lý nhóm & Thành viên

**Chức năng chính:**

* Tạo nhóm mới
* Mời thành viên qua email hoặc mã mời
* Phân công vai trò trong nhóm (leader, member)
* Quản lý danh sách thành viên
* Xóa hoặc rời nhóm

**Dữ liệu xử lý:**

* Thông tin nhóm (groupId, tên nhóm, mô tả, ngày tạo)
* Danh sách thành viên (userId, vai trò trong nhóm)

**Tương tác:**

* Liên kết với **Activity Service** để ghi nhận hoạt động nhóm
* Liên kết với **Document Service** để quản lý tài liệu theo nhóm

3. **Document Service** – Quản lý tài liệu & Kết nối MCP

**Chức năng chính:**

* Upload tài liệu trực tiếp (PDF, DOCX, TXT…)
* Liên kết tài liệu từ Google Drive, OneDrive, hệ thống nội bộ qua MCP
* Lưu metadata tài liệu (tên, nguồn, người tải lên)
* Trích xuất nội dung tài liệu qua MCP
* Cập nhật hoặc xóa tài liệu

**Dữ liệu xử lý:**

* Thông tin tài liệu (documentId, tiêu đề, nguồn, link, nội dung trích xuất)
* Metadata (ngày tạo, người tạo, nhóm liên kết)

**Tương tác:**

* Gửi yêu cầu đến **MCP Server** để lấy nội dung
* Cung cấp dữ liệu cho **AI Service** để tóm tắt hoặc tạo câu hỏi

4. **Activity Service** – Ghi nhận hoạt động nhóm

**Chức năng chính:**

* Lưu lại các hành động của thành viên (tải tài liệu, chỉnh sửa, hoàn thành nhiệm vụ)
* Ghi nhận các sự kiện quan trọng (họp nhóm, deadline)
* Tạo timeline hoạt động nhóm

**Dữ liệu xử lý:**

* Nhật ký hoạt động (activityId, loại hoạt động, người thực hiện, thời gian, mô tả)
* Liên kết với nhóm và tài liệu liên quan

**Tương tác:**

* Cung cấp dữ liệu cho **AI Service** để tóm tắt tiến độ nhóm
* Hiển thị trên dashboard nhóm

5. **AI Service** – Xử lý ngôn ngữ tự nhiên

**Chức năng chính:**

* Nhận truy vấn từ người dùng
* Lấy ngữ cảnh từ **Document Service** và **Activity Service**
* Gửi yêu cầu đến LLM (GPT-4, Claude…) để sinh câu trả lời
* Tóm tắt tài liệu, tạo câu hỏi, phân tích tiến độ nhóm
* Trả lời các câu hỏi như “Ai chưa hoàn thành nhiệm vụ?”

**Dữ liệu xử lý:**

* Truy vấn người dùng
* Ngữ cảnh từ tài liệu và hoạt động nhóm
* Phản hồi từ LLM

**Tương tác:**

* Liên kết với **Memory Service** để lưu câu hỏi và câu trả lời
* Gọi **MCP Server** để lấy dữ liệu cần thiết

6. **Memory Service** – Bộ nhớ dài hạn

**Chức năng chính:**

* Lưu lại toàn bộ câu hỏi và câu trả lời của AI
* Gắn nhãn nguồn trả lời (từ tài liệu, từ AI, từ hoạt động nhóm)
* Cho phép AI “ghi nhớ” thông tin để trả lời nhanh hơn lần sau
* Hỗ trợ tìm kiếm lại câu trả lời cũ

**Dữ liệu xử lý:**

* Lịch sử hội thoại (question, answer, source, timestamp)
* Liên kết với người dùng và nhóm

**Tương tác:**

* Cung cấp dữ liệu cho **AI Service** khi cần ngữ cảnh lịch sử
* Cho phép người dùng xem lại câu trả lời cũ

🔄 **Luồng tương tác giữa các service**

1. Người dùng gửi yêu cầu qua **Frontend**
2. **Auth Service** xác thực quyền truy cập
3. **Group Service** xác định nhóm và thành viên liên quan
4. **Document Service** hoặc **Activity Service** cung cấp dữ liệu ngữ cảnh
5. **AI Service** xử lý truy vấn, gọi LLM nếu cần
6. **Memory Service** lưu kết quả để dùng cho lần sau
7. Phản hồi trả về **Frontend** hiển thị cho người dùng

**Kịch bản trải nghiệm người dùng**

**Nhân vật**

* **Người dùng:** Lan – sinh viên năm 3, trưởng nhóm dự án môn Hệ thống thông tin.
* **Mục tiêu:** Hoàn thành báo cáo nhóm đúng hạn, quản lý tiến độ và tài liệu hiệu quả.

**Bước 1: Đăng nhập & Tạo nhóm**

* Lan truy cập ứng dụng web, đăng nhập bằng email trường.
* Chọn "Tạo nhóm mới", đặt tên "Nhóm 5 – Hệ thống thông tin".
* Mời 4 thành viên khác qua email.
* Trải nghiệm: Giao diện đơn giản, chỉ vài click là nhóm đã sẵn sàng.

**Bước 2: Kết nối tài liệu qua MCP**

* Lan liên kết thư mục Google Drive của nhóm vào hệ thống.
* MCP tự động liệt kê các file liên quan: “Chương 1 – Giới thiệu”, “Chương 2 – Phân tích yêu cầu”...
* Trải nghiệm: Không cần tải file thủ công, mọi tài liệu nhóm đều đồng bộ tức thì.

**Bước 3: Phân công công việc**

* Lan tạo danh sách nhiệm vụ:
* “Viết phần giới thiệu” → giao cho Minh, deadline 10/09.
* “Thiết kế CSDL” → giao cho Huy, deadline 12/09.
* Thành viên nhận thông báo ngay trên dashboard.
* Trải nghiệm: Mọi người biết rõ mình phải làm gì và khi nào.

**Bước 4: Ghi nhận hoạt động nhóm**

* Minh upload bản nháp phần giới thiệu → hệ thống ghi lại trong Activity Timeline.
* Huy chỉnh sửa sơ đồ CSDL → cũng được ghi nhận.
* Trải nghiệm: Lan có thể xem toàn bộ tiến độ nhóm trong một dòng thời gian rõ ràng.

**Bước 5: Hỏi AI về tiến độ**

* Lan gõ: “Nhóm mình còn nhiệm vụ nào chưa xong?”
* AI Service lấy dữ liệu từ Activity Service và Document Service, trả lời:
* “Còn 2 nhiệm vụ chưa hoàn thành: Viết phần kết luận (Nam) và Vẽ sơ đồ kiến trúc (Huy).”
* Trải nghiệm: Lan không cần hỏi từng người, AI đã tổng hợp sẵn.

**Bước 6: Tóm tắt tài liệu bằng AI**

* Lan mở file “Chương 3 – Kiến trúc hệ thống” và yêu cầu AI tóm tắt.
* AI trả về 5 ý chính + 10 câu hỏi trắc nghiệm để nhóm ôn tập.
* Trải nghiệm: Tiết kiệm thời gian đọc, dễ dàng nắm nội dung chính.

**Bước 7: Lưu và tra cứu lịch sử hội thoại**

* Tuần sau, Lan muốn xem lại câu trả lời “Các bước triển khai MCP”.
* Mở Memory Service → tìm kiếm từ khóa “MCP” → hiển thị câu trả lời cũ.
* Trải nghiệm: Không mất công hỏi lại, mọi thông tin đều được lưu.

**Bước 8: Hoàn thành dự án**

* Tất cả nhiệm vụ được đánh dấu hoàn thành.
* AI tạo báo cáo tóm tắt tiến độ nhóm để nộp kèm sản phẩm.
* Trải nghiệm: Lan dễ dàng tổng hợp kết quả và đảm bảo không bỏ sót phần nào.

📌 **Điểm nhấn trong trải nghiệm**

* **Nhanh: Kết nối tài liệu tự động qua MCP.**
* **Thông minh: AI trả lời theo ngữ cảnh nhóm.**
* **Nhớ lâu: Lưu toàn bộ lịch sử để tra cứu.**
* **Minh bạch: Timeline hoạt động rõ ràng, ai làm gì đều biết.**

Cong-viec-nhom-AI/

│

├── services/

│ ├── auth-service/

│ │ ├── controllers/

│ │ │ └── authController.js

│ │ ├── models/

│ │ │ └── User.js

│ │ ├── routes/

│ │ │ └── authRoutes.js

│ │ ├── middleware/

│ │ │ └── authMiddleware.js

│ │ ├── config/

│ │ │ └── db.js

│ │ └── server.js

│ │

│ ├── group-service/

│ │ ├── controllers/

│ │ │ └── groupController.js

│ │ ├── models/

│ │ │ └── Group.js

│ │ ├── routes/

│ │ │ └── groupRoutes.js

│ │ └── server.js

│ │

│ ├── document-service/

│ │ ├── controllers/

│ │ │ └── documentController.js

│ │ ├── models/

│ │ │ └── Document.js

│ │ ├── routes/

│ │ │ └── documentRoutes.js

│ │ └── server.js

│ │

│ ├── activity-service/

│ │ ├── controllers/

│ │ │ └── activityController.js

│ │ ├── models/

│ │ │ └── Activity.js

│ │ ├── routes/

│ │ │ └── activityRoutes.js

│ │ └── server.js

│ │

│ ├── ai-service/

│ │ ├── controllers/

│ │ │ └── aiController.js

│ │ ├── utils/

│ │ │ └── llmClient.js # Kết nối GPT-4/Claude

│ │ ├── routes/

│ │ │ └── aiRoutes.js

│ │ └── server.js

│ │

│ └── memory-service/

│ ├── controllers/

│ │ └── memoryController.js

│ ├── models/

│ │ └── Memory.js

│ ├── routes/

│ │ └── memoryRoutes.js

│ └── server.js

│

├── gateway/

│ ├── routes/

│ │ └── index.js # Định tuyến đến các service

│ ├── middleware/

│ │ └── authMiddleware.js

│ └── server.js # API Gateway

│

├── docker-compose.yml # Chạy nhiều service cùng lúc

├── .env

└── package.json