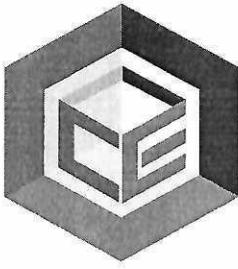


LƯU Ý KHI LÀM CHƯƠNG TRÌNH:

- DN không tự ý điều chỉnh/ xóa bớt thông tin trong mẫu chương trình, chỉ thêm thông tin trong phần ba chấm (...).
- Chương trình cần phải được Khoa duyệt/ công bố trước khi nhận SV đến thực tập.
- DN làm chương trình gửi file dạng WORD (.docx) về Khoa qua email trước. Sau khi được Khoa duyệt cả về hình thức lẫn nội dung, DN mới in - ký - đóng dấu - scan màu - gửi file scan màu về Khoa qua email người đại diện. Bản giấy, DN giữ lại gửi về Khoa qua đường bưu điện cùng kết quả thực tập/ hồ sơ liên quan khác vào cuối kỳ.
- Người đại diện Khoa nhận và xử lý thông tin: cô Thu Trang – thutrangcse@hcmut.edu.vn

THÔNG TIN VỀ DOANH NGHIỆP (DN):

	CÔNG TY CỔ PHẦN KỸ THUẬT ĐIỆN TOÁN BÁCH KHOA Địa chỉ trụ sở: 174 Đường số 1, Phường Bình Trị Đông B, Quận Bình Tân, Thành phố Hồ Chí Minh. Văn phòng làm việc: 437/16 Hoàng Văn Thụ, phường 4, quận Tân Bình, TP. Hồ Chí Minh Mã số thuế: 0314042642 Điện thoại: 0909 935 920 (Bảo Châu - HR) Email: hr@dientoan.vn Web link: http://dientoan.vn/
---	--

CHƯƠNG TRÌNH THỰC TẬP TỐT NGHIỆP (TTTN)/ THỰC TẬP NGOÀI TRƯỜNG (TTNT) DÀNH RIÊNG CHO SINH VIÊN KHOA KH&KT MÁY TÍNH – TRƯỜNG ĐH BÁCH KHOA – ĐHQG TP.HCM HỌC KỲ 3/2024-2025 (HK243) (Thời gian thực tập thực tế từ 16/06-15/08/2025)

I. NỘI DUNG:

1. Giới thiệu sơ bộ về DN:

Công ty Cổ phần Kỹ thuật Điện Toán Bách Khoa được thành lập vào tháng 10 năm 2016 bởi một nhóm nhà khoa học trẻ là những kỹ sư/thạc sĩ/tiến sĩ và cũng là những cựu sinh viên của trường Đại học Bách Khoa, nhằm nghiên cứu những giải pháp và ứng dụng các công nghệ tiên tiến vào cuộc sống, với hoài bão gop phần tích cực vào công cuộc chuyển đổi, công nghiệp 4.0, hiện đại hóa quê hương Việt Nam.

Trong suốt quá trình hoạt động, công ty luôn nhạy bén, đi đầu trong việc nhắm bắt, giới thiệu, cung cấp thiết bị hiện đại, dịch vụ kỹ thuật cao, giải pháp tích hợp và chuyển giao công nghệ, công ty có lĩnh hoạt động vừa đa dạng, vừa chuyên sâu, được đối tác và khách hàng tín nhiệm.

Công ty Cổ phần Kỹ thuật Điện toán Bách Khoa hiện đang là một trong những doanh nghiệp đang đi đầu trong lĩnh vực công nghệ về trí tuệ nhân tạo. Với đội ngũ nhân viên có trình độ cao, nhiệt tình, giàu kinh nghiệm, trẻ tuổi và sáng tạo trong công việc. Bên cạnh đó bộ máy tổ chức năng động và chuyên nghiệp, nguồn tài chính lành mạnh, các thành viên của công ty luôn được các đối tác, khách hàng trong và ngoài nước tin tưởng, đánh giá cao. Không những hoạt động trong lĩnh vực trí tuệ nhân tạo, doanh nghiệp còn cung cấp các giải pháp tổng thể, triển khai kinh doanh và cung cấp dịch vụ chuyên nghiệp về công nghệ với trọng tâm là mang đến sự hài lòng nhất cho khách hàng.

2. Chương trình:

Ngoài kiến thức chuyên môn về ngành Khoa học và Kỹ thuật Máy Tính - Công nghệ Thông tin, chương trình **đảm bảo/ cam kết** trang bị cho sinh viên (SV) đầy đủ kỹ năng sau:

(DN sẽ chấm điểm kỳ thực tập cho SV theo tiêu chí trong bảng dưới đây – mẫu bảng điểm đính kèm - file D4).

Số thứ tự	Tiêu chí	Điểm tối đa
1	Kỹ năng chuyên môn: Mức độ hoàn thành công việc.	20
2	Khả năng làm việc theo nhóm:	30
a	Khả năng xác định được vai trò, vị trí và công việc của mình trong nhóm. Xác định được quy trình làm việc trong nhóm một cách rõ ràng.	10
b	Khả năng cộng tác với các thành viên khác trong nhóm. Khả năng trao đổi với các thành viên trong nhóm để phối hợp thực hiện các nhiệm vụ chung và nhiệm vụ của bản thân. Khả năng dung hòa được các hoạt động của mình với tiến độ chung. Khả năng giúp đỡ các thành viên khác nếu có thể để cùng đảm bảo hoàn thành kế hoạch nhóm đúng tiến độ.	15
c	Mức độ tích cực đóng góp cho các hoạt động của nhóm. SV phải đóng góp ý kiến tích cực trong các buổi họp nhóm nhằm giải quyết vấn đề của nhóm cũng như giúp nhóm phát triển tốt hơn.	5
3	Kỹ năng giao tiếp:	35
a	Mức độ tham gia các hoạt động training về giao tiếp của chương trình. SV phải tham gia đầy đủ các buổi training được tổ chức.	5
b	Khả năng trình bày (presentation skill): Khả năng chuẩn bị, tổ chức một buổi thuyết trình theo yêu cầu bao gồm việc chuẩn bị tư liệu, bài trình bày, phương tiện, kế hoạch,...	5

c	Khả năng sử dụng kỹ năng thuyết trình.	5
d	Khả năng sử dụng các trang thiết bị phục vụ cho bài presentation như máy tính, phần mềm, máy chiếu,...	5
e	Khả năng sử dụng kỹ năng viết báo cáo. Khả năng nắm bắt các phương thức trình bày các dạng văn bản phổ biến như: thư, thư điện tử, đề nghị, báo cáo,...	15
4	Khả năng hòa nhập với môi trường doanh nghiệp:	15
a	Mức độ tuân thủ kỷ luật làm việc về: giờ giấc, nội quy, tác phong, mức độ chấp hành yêu cầu của cấp trên.	5
b	Mức độ hòa nhập với các hoạt động chung của DN, khả năng hòa nhập với văn hóa DN. SV phải tham gia một cách tích cực và hòa đồng với các hoạt động chung.	5
c	Mức độ hiểu biết về cách tổ chức hoạt động của DN: văn hóa tổ chức, giao tiếp & sinh hoạt của DN, mức độ hòa nhập với văn hóa chung của DN.	5
Tổng điểm:		100

- **Số lượng suất thực tập: 06-09**
- Đối tượng SV (yêu cầu **tối thiểu/ đặc biệt** dành cho SV đủ chuẩn đăng ký/ nộp hồ sơ ứng tuyển - nếu có):
 -
 - Danh sách SV đã trúng tuyển (nếu đã có):

(Nếu danh sách SV dưới đây đã đầy đủ thì DN không cần làm/ nộp Khoa form D3 - công văn xác nhận kết quả xét tuyển)

Số	Mã số sinh viên	Họ lót	Tên	Ghi chú
1	
2	
...	

- Tổng thời gian làm việc của đợt thực tập: từ ngày **16/06-15/08/2025 (tối thiểu 08 tuần)**. Khoa đưa ra thời gian này là lý tưởng so với chương trình/ lộ trình học của SV tại Khoa/ Trường. Nếu DN và SV làm việc với nhau khác thời gian trên sẽ trên tinh thần thuận riêng 02 bên DN và SV -

- không đưa vào chương trình - Khoa sẽ không chịu trách nhiệm hỗ trợ xử lý các vấn đề liên quan ngoài thời gian quy định.*
- Thời gian làm việc trong ngày: Theo thoả thuận. **Tối thiểu SV sẽ được sắp xếp ngoài làm việc trực tiếp tại trụ sở DN 2,5 ngày trong một tuần.**
 - Địa điểm làm việc (SV cần được DN sắp xếp chỗ ngồi làm việc tại DN, để SV có cơ hội hấp thu văn hóa DN trong quá trình thực tập):
 - o Văn phòng làm việc tại 437/16 Hoàng Văn Thụ, phường 4, quận Tân Bình, TP. Hồ Chí Minh.
 - o Trong quá trình thực tập, sinh viên được công ty sắp xếp vị trí làm việc rộng rãi, thoáng mát.
 - o Môi trường làm việc năng động, chuyên nghiệp, sáng tạo.
 - o Hệ thống an ninh bảo vệ của Công ty được hoạt động liên tục từ 8h - 18h đảm bảo tài sản cho các bạn sinh viên thực tập.
 - **Ưu đãi (nếu có):**
 - o Được đào tạo công nghệ, kỹ năng làm việc cơ bản và kỹ năng mềm.
 - o Được học hỏi tiếp xúc, làm việc với các anh chị nhiều kinh nghiệm trong lĩnh vực.
 - o Được định hướng về công việc và những chia sẻ của BLĐ Công ty.
 - o Sau khi hoàn thành chương trình thực tập, sinh viên thực tập có cơ hội được tiếp nhận làm việc chính thức tại Công ty Cổ phần Kỹ thuật Điện Toán Bách Khoa.
 - o Luôn được hỗ trợ cà phê, trà, bánh kẹo trong suốt quá trình thực tập.
 - o Hàng tháng công ty đều tổ chức party cho CB - NV trong Công ty, vậy nên các bạn sinh viên thực tập sẽ cùng tham gia.
 - Thông tin thêm (nếu có):
 - Khối lượng/ nội dung công việc cụ thể:
(DN cần nêu rõ tên/ nội dung đề tài, ngôn ngữ lập trình/ framework mà SV được tham gia/ sử dụng).

Đề tài 1 (2-3 sinh viên): Xây dựng Mini Web App trò chuyện với AI để truy vấn dữ liệu với RabbitMQ

Tuần	Nội dung công việc
1	<p>Nghiên cứu và Thiết lập</p> <ul style="list-style-type: none"> • Mục tiêu: Hiểu tech stack, API LLM, và thiết lập môi trường. • Nhiệm vụ: <ul style="list-style-type: none"> o Đọc tài liệu API LLM (ví dụ: Grok, OpenAI Chat Completions). o Tìm hiểu NestJS (API REST), PostgreSQL (truy vấn SQL), RabbitMQ (hàng đợi tin nhắn). o Nghiên cứu ReactJS (Vite), Tailwind CSS, và React Query (gọi API). o Thiết lập môi trường: <ul style="list-style-type: none"> ▪ Cài Node.js, Docker (cho PostgreSQL, RabbitMQ). ▪ Khởi tạo dự án NestJS (TypeScript) và ReactJS (Vite).

	<ul style="list-style-type: none"> ▪ Cài đặt thư viện như axios (cho gọi API), @nestjs/typeorm, amqplib, react-query. ◦ Tạo kho lưu trữ GitHub để quản lý mã. ◦ Thiết lập bảng mẫu trong PostgreSQL (ví dụ: bảng sales với cột product_id, quantity, revenue, date). • Kết quả: Đề xuất dự án (1 trang) nêu trường hợp sử dụng, tech stack, và lộ trình.
2	<p>Khám phá API LLM và PostgreSQL</p> <ul style="list-style-type: none"> • Mục tiêu: Thủ nghiệm API LLM và truy vấn PostgreSQL. • Nhiệm vụ: <ul style="list-style-type: none"> ◦ Sử dụng khóa API LLM để thử gọi API (ví dụ: gửi câu hỏi “Tổng doanh số?” và nhận câu SQL như SELECT SUM(revenue) FROM sales) bằng axios trong NestJS. ◦ Viết mã TypeScript trong NestJS để gọi API LLM, định dạng câu lệnh: “Tạo câu SQL cho câu hỏi: {câu hỏi} dựa trên bảng sales (product_id, quantity, revenue, date)”. ◦ Kết nối NestJS với PostgreSQL (TypeORM) và thử truy vấn bảng sales. ◦ Kiểm tra RabbitMQ bằng cách gửi tin nhắn thử (dùng amqplib). • Kết quả: Mã TypeScript với gọi API LLM, truy vấn PostgreSQL mẫu, và tin nhắn RabbitMQ thử.
3	<p>Phát triển Backend - Tích hợp LLM và PostgreSQL</p> <ul style="list-style-type: none"> • Mục tiêu: Xây dựng backend để xử lý câu hỏi và truy vấn dữ liệu. • Nhiệm vụ: <ul style="list-style-type: none"> ◦ Tạo API REST trong NestJS với điểm cuối /query nhận câu hỏi người dùng. ◦ Gọi API LLM bằng axios để chuyển câu hỏi thành câu SQL (ví dụ: “Tổng số sản phẩm bán được?” → SELECT SUM(quantity) FROM sales). ◦ Thực thi câu SQL trên PostgreSQL (TypeORM) và trả kết quả. ◦ Lưu khóa API LLM an toàn bằng biến môi trường (tệp .env). ◦ Kiểm tra điểm cuối bằng Postman với câu hỏi mẫu. • Kết quả: Backend NestJS với điểm cuối /query, tích hợp LLM và PostgreSQL, kèm kết quả kiểm tra.
4	<p>Phát triển Backend - Tích hợp RabbitMQ</p> <ul style="list-style-type: none"> • Mục tiêu: Thêm RabbitMQ để xử lý hàng đợi câu hỏi. • Nhiệm vụ: <ul style="list-style-type: none"> ◦ Cấu hình RabbitMQ trong NestJS (sử dụng amqplib hoặc @nestjs/microservices). ◦ Tạo producer gửi câu hỏi người dùng đến queue RabbitMQ (ví dụ: user_queries). ◦ Tạo consumer nhận câu hỏi, gọi LLM, thực thi SQL, và trả kết quả về queue khác (ví dụ: query_results). ◦ Xử lý lỗi cơ bản (ví dụ: API LLM thất bại, SQL không hợp lệ). ◦ Kiểm tra tích hợp RabbitMQ bằng câu hỏi mẫu. • Kết quả: Backend NestJS tích hợp RabbitMQ, LLM, và PostgreSQL, kèm kết quả kiểm tra.
5	<p>Phát triển Frontend</p> <ul style="list-style-type: none"> • Mục tiêu: Xây dựng giao diện trò chuyện đơn giản. • Nhiệm vụ:

	<ul style="list-style-type: none"> ○ Tạo ứng dụng ReactJS với Vite, sử dụng Tailwind CSS để tạo giao diện. ○ Thiết kế giao diện gồm: <ul style="list-style-type: none"> ▪ Ô nhập câu hỏi (text input). ▪ Khu vực hiển thị kết quả (text output). ○ Sử dụng React Query để gọi API /query từ backend. ○ Hiển thị trạng thái (ví dụ: “Đang xử lý”, “Kết quả: {kết quả}”). ○ Kiểm tra giao diện với câu hỏi mẫu (ví dụ: “Doanh số tháng 10?”). <p>• Kết quả: Frontend ReactJS với giao diện trò chuyện cơ bản.</p>
6	<p>Kiểm thử Tích hợp</p> <ul style="list-style-type: none"> • Mục tiêu: Kiểm tra toàn bộ ứng dụng. • Nhiệm vụ: <ul style="list-style-type: none"> ○ Kiểm tra end-to-end: nhập câu hỏi → gửi qua RabbitMQ → LLM tạo SQL → truy vấn PostgreSQL → hiển thị kết quả trên giao diện. ○ Kiểm tra các trường hợp đơn giản (ví dụ: câu hỏi hợp lệ, API lỗi). ○ Sửa lỗi (ví dụ: kết quả không hiển thị, RabbitMQ không gửi). ○ Cải thiện trải nghiệm (ví dụ: thêm thông báo “Đang tải”). <p>• Kết quả: Ứng dụng tích hợp hoàn chỉnh với báo cáo kiểm tra.</p>
7	<p>Triển khai</p> <ul style="list-style-type: none"> • Mục tiêu: Triển khai ứng dụng lên nền tảng đám mây. • Nhiệm vụ: <ul style="list-style-type: none"> ○ Sử dụng nền tảng miễn phí/chi phí thấp (ví dụ: Heroku, Render). ○ Triển khai backend (NestJS, PostgreSQL, RabbitMQ) và frontend (ReactJS). ○ Cấu hình biến môi trường cho khóa API LLM và kết nối PostgreSQL. ○ Kiểm tra ứng dụng trực tuyến với câu hỏi mẫu. <p>• Kết quả: URL ứng dụng trực tiếp với tính năng trò chuyện AI.</p>
8	<p>Tài liệu và Trình bày</p> <ul style="list-style-type: none"> • Mục tiêu: Lập tài liệu và trình bày công việc. • Nhiệm vụ: <ul style="list-style-type: none"> ○ Viết báo cáo dự án (1-2 trang) bao gồm: <ul style="list-style-type: none"> ▪ Tổng quan: Mini web app trò chuyện với AI. ▪ Chi tiết tích hợp (LLM, PostgreSQL, RabbitMQ). ▪ Cấu trúc hệ thống (frontend, backend). ▪ Thách thức và giải pháp (ví dụ: tạo câu SQL chính xác). ▪ Ý tưởng cải tiến (ví dụ: hỗ trợ Google Sheets). ○ Tạo bài thuyết trình 3 phút tóm tắt dự án. ○ Ghi video demo (1-2 phút) thể hiện tính năng. <p>• Kết quả: Báo cáo, slide thuyết trình, và video demo.</p>

Đề tài 2 (2-3 sinh viên): Xây dựng Mini Web App đồng bộ Google Sheets với Shopee sử dụng RabbitMQ

Tuần	Nội dung công việc
1	<p>Nghiên cứu và Thiết lập</p> <ul style="list-style-type: none"> • Mục tiêu: Hiểu API Shopee, Google Sheets, RabbitMQ, và thiết lập môi trường. • Nhiệm vụ:

	<ul style="list-style-type: none"> ○ Đọc tài liệu API Shopee (Product API để cập nhật tồn kho, Order API để lấy đơn hàng). ○ Đọc tài liệu Google Sheets API (đọc/ghi dữ liệu từ sheet). ○ Tìm hiểu NestJS (API REST), PostgreSQL (truy vấn SQL), RabbitMQ (hàng đợi tin nhắn). ○ Nghiên cứu ReactJS (Vite), Tailwind CSS, React Query (gọi API). ○ Thiết lập môi trường: <ul style="list-style-type: none"> ▪ Cài Node.js, Docker (cho PostgreSQL, RabbitMQ). ▪ Khởi tạo dự án NestJS (TypeScript) và ReactJS (Vite). ▪ Cài đặt thư viện như axios, @nestjs/typeorm, amqplib, react-query. ○ Tạo kho lưu trữ GitHub để quản lý mã. ○ Thiết lập Google Sheet mẫu (sheet “Products” với cột product_id, name, stock; sheet “Orders” với cột order_id, status, date). ○ Thiết lập bảng PostgreSQL mẫu (bảng sync_logs với cột id, timestamp, status). <ul style="list-style-type: none"> ● Kết quả: Đề xuất dự án (1 trang) nêu trường hợp sử dụng, tech stack, và lộ trình.
2	<p>Khám phá API Shopee, Google Sheets, và RabbitMQ</p> <ul style="list-style-type: none"> ● Mục tiêu: Thủ nghiệm API Shopee, Google Sheets, PostgreSQL, và RabbitMQ. ● Nhiệm vụ: <ul style="list-style-type: none"> ○ Sử dụng khóa API Shopee để thử gọi API (ví dụ: cập nhật tồn kho, lấy đơn hàng) bằng axios trong NestJS. ○ Sử dụng Google Sheets API để đọc/ghi dữ liệu (ví dụ: đọc stock từ sheet “Products”, ghi order_id vào sheet “Orders”) bằng axios. ○ Kết nối NestJS với PostgreSQL (TypeORM) và thử ghi nhật ký đồng bộ vào bảng sync_logs. ○ Kiểm tra RabbitMQ bằng cách gửi và nhận tin nhắn thử (dùng amqplib, ví dụ: queue sync_requests). ○ Ánh xạ dữ liệu cơ bản (ví dụ: product_id từ Google Sheet với product_id trên Shopee). ● Kết quả: Mã TypeScript với gọi API mẫu, truy vấn PostgreSQL, tin nhắn RabbitMQ thử, và ánh xạ dữ liệu.
3	<p>Phát triển Backend - Tích hợp Shopee, Google Sheets, và RabbitMQ cho Đồng bộ Sản phẩm</p> <ul style="list-style-type: none"> ● Mục tiêu: Xây dựng backend để đồng bộ sản phẩm qua RabbitMQ. ● Nhiệm vụ: <ul style="list-style-type: none"> ○ Tạo API REST trong NestJS với điểm cuối /sync_products nhận yêu cầu đồng bộ sản phẩm. ○ Cấu hình RabbitMQ: <ul style="list-style-type: none"> ▪ Tạo producer gửi yêu cầu đồng bộ sản phẩm (dữ liệu từ sheet “Products”) đến queue product_sync_requests. ▪ Tạo consumer nhận yêu cầu, đọc dữ liệu từ Google Sheets (Google Sheets API), gửi đến Shopee (Product API), và trả kết quả về queue product_sync_results. ○ Ghi nhật ký đồng bộ vào bảng sync_logs (TypeORM). ○ Lưu khóa API Shopee và Google Sheets an toàn bằng biến môi trường (tệp .env). ○ Kiểm tra điểm cuối bằng Postman với dữ liệu sản phẩm mẫu. ● Kết quả: Backend NestJS với điểm cuối /sync_products, tích hợp Shopee, Google Sheets, và RabbitMQ, kèm kết quả kiểm tra.
4	<p>Phát triển Backend - Tích hợp Đồng bộ Đơn hàng qua RabbitMQ</p> <ul style="list-style-type: none"> ● Mục tiêu: Xây dựng backend để đồng bộ đơn hàng qua RabbitMQ. ● Nhiệm vụ:

	<ul style="list-style-type: none"> ○ Tạo điểm cuối /sync_orders nhận yêu cầu đồng bộ đơn hàng. ○ Cấu hình RabbitMQ: <ul style="list-style-type: none"> ▪ Tạo producer gửi yêu cầu đồng bộ đơn hàng đến queue order_sync_requests. ▪ Tạo consumer nhận yêu cầu, gọi API Shopee (Order API) để lấy dữ liệu đơn hàng, ghi vào sheet “Orders” (Google Sheets API), và trả kết quả về queue order_sync_results. ○ Ghi nhật ký đồng bộ vào bảng sync_logs (TypeORM). ○ Xử lý lỗi cơ bản (ví dụ: API Shopee thất bại, sheet không tồn tại). ○ Kiểm tra điểm cuối bằng Postman với dữ liệu đơn hàng mẫu. <ul style="list-style-type: none"> • Kết quả: Backend NestJS với điểm cuối /sync_orders, tích hợp đầy đủ Shopee, Google Sheets, PostgreSQL, và RabbitMQ.
5	<p>Phát triển Frontend</p> <ul style="list-style-type: none"> • Mục tiêu: Xây dựng giao diện đồng bộ đơn giản. • Nhiệm vụ: <ul style="list-style-type: none"> ○ Tạo ứng dụng ReactJS với Vite, sử dụng Tailwind CSS để tạo giao diện. ○ Thiết kế giao diện gồm: <ul style="list-style-type: none"> ▪ Nút “Đồng bộ sản phẩm” để gọi /sync_products. ▪ Nút “Đồng bộ đơn hàng” để gọi /sync_orders. ▪ Khu vực hiển thị trạng thái (ví dụ: “Đồng bộ thành công”, “Lỗi”). ○ Sử dụng React Query để gọi API /sync_products và /sync_orders. ○ Tạo kiểu giao diện trực quan, responsive. ○ Kiểm tra giao diện với dữ liệu mẫu. • Kết quả: Frontend ReactJS với giao diện đồng bộ cơ bản.
6	<p>Kiểm thử Tích hợp</p> <ul style="list-style-type: none"> • Mục tiêu: Kiểm tra toàn bộ ứng dụng. • Nhiệm vụ: <ul style="list-style-type: none"> ○ Kiểm tra end-to-end: <ul style="list-style-type: none"> ▪ Đồng bộ sản phẩm: Google Sheet → RabbitMQ → Shopee → ghi nhật ký. ▪ Đồng bộ đơn hàng: Shopee → RabbitMQ → Google Sheet → ghi nhật ký. ○ Kiểm tra các trường hợp (ví dụ: dữ liệu hợp lệ, API lỗi, RabbitMQ không gửi). ○ Sửa lỗi (ví dụ: trạng thái không hiển thị, đồng bộ thất bại). ○ Cải thiện trải nghiệm (ví dụ: thêm thông báo “Đang đồng bộ”). • Kết quả: Ứng dụng tích hợp hoàn chỉnh với báo cáo kiểm tra.
7	<p>Triển khai</p> <ul style="list-style-type: none"> • Mục tiêu: Triển khai ứng dụng lên nền tảng đám mây. • Nhiệm vụ: <ul style="list-style-type: none"> ○ Sử dụng nền tảng miễn phí/chi phí thấp (ví dụ: Heroku, Render). ○ Triển khai backend (NestJS, PostgreSQL, RabbitMQ) và frontend (ReactJS). ○ Cấu hình biến môi trường cho khóa API Shopee và Google Sheets. ○ Kiểm tra ứng dụng trực tuyến với dữ liệu mẫu. • Kết quả: URL ứng dụng trực tiếp với tính năng đồng bộ.
8	<p>Tài liệu và Trình bày</p> <ul style="list-style-type: none"> • Mục tiêu: Lập tài liệu và trình bày công việc. • Nhiệm vụ: <ul style="list-style-type: none"> ○ Viết báo cáo dự án (1-2 trang) bao gồm:

	<ul style="list-style-type: none"> ▪ Tổng quan: Mini web app đồng bộ Google Sheets và Shopee với RabbitMQ. ▪ Chi tiết tích hợp (Shopee, Google Sheets, PostgreSQL, RabbitMQ). ▪ Cấu trúc hệ thống (frontend, backend). ▪ Thách thức và giải pháp (ví dụ: ánh xạ dữ liệu, xử lý lỗi RabbitMQ). ▪ Ý tưởng cải tiến (ví dụ: đồng bộ tự động theo lịch). <ul style="list-style-type: none"> ◦ Tạo bài thuyết trình 3 phút tóm tắt dự án. ◦ Ghi video demo (1-2 phút) thể hiện tính năng. <p>• Kết quả: Báo cáo, slide thuyết trình, và video demo.</p>
--	---

Đề tài 3 (2-3 sinh viên): Phát triển AI Agent tìm kiếm Database

Tuần	Nội dung công việc
1	<p>Nghiên cứu và lựa chọn công cụ No-code/Low-code</p> <ul style="list-style-type: none"> • Mục tiêu: Tìm hiểu và lựa chọn công cụ no-code/low-code phù hợp với dự án • Nhiệm vụ: <ul style="list-style-type: none"> ◦ Nghiên cứu các nền tảng no-code/low-code hỗ trợ kết nối LLM API: <ul style="list-style-type: none"> ▪ Flowise/Langflow ▪ n8n/Make (Integromat) ▪ Bubble.io với plugin AI ▪ Microsoft Promptflow ▪ Zapier với tích hợp AI ◦ So sánh các công cụ dựa trên: <ul style="list-style-type: none"> ▪ Khả năng kết nối database ▪ Tích hợp các dịch vụ LLM (OpenAI, Claude, Gemini) ▪ Tính năng bảo mật và kiểm soát chi phí ▪ Độ phức tạp khi triển khai ◦ Học cách sử dụng 2-3 công cụ triển vọng nhất ◦ Tạo các workflow đơn giản để thử nghiệm khả thi • Kết quả: <ul style="list-style-type: none"> ◦ Báo cáo so sánh các công cụ no-code/low-code ◦ Quyết định lựa chọn công cụ phù hợp nhất ◦ Demo workflow đơn giản thể hiện khả năng kết nối LLM và database
2	<p>Thiết lập kết nối Database và LLM API</p> <ul style="list-style-type: none"> • Mục tiêu: Kết nối thành công database và dịch vụ LLM thông qua công cụ đã chọn • Nhiệm vụ: <ul style="list-style-type: none"> ◦ Thiết lập kết nối với database: <ul style="list-style-type: none"> ▪ Cấu hình các tham số kết nối ▪ Xác định schema và các bảng cần truy xuất ▪ Tạo các query mẫu để truy xuất dữ liệu ◦ Thiết lập kết nối với LLM API: <ul style="list-style-type: none"> ▪ Đăng ký và lấy API key ▪ Cấu hình thông số cho LLM (temperature, max_tokens) ▪ Thủ nghiệm prompt đơn giản ◦ Tạo workflow cơ bản để xử lý: <ul style="list-style-type: none"> ▪ Nhận input từ người dùng ▪ Truy vấn database ▪ Gửi prompt đến LLM

	<ul style="list-style-type: none"> ▪ Hiển thị kết quả <ul style="list-style-type: none"> • Kết quả: <ul style="list-style-type: none"> ○ Workflow kết nối được với database ○ Workflow kết nối được với LLM API ○ Báo cáo thử nghiệm kết nối và các vấn đề gặp phải ○ Pipeline cơ bản từ input đến output
3	<p>Áp dụng Model Context Protocol (MCP) và cơ chế bảo mật</p> <ul style="list-style-type: none"> • Mục tiêu: Tích hợp MCP và xây dựng các biện pháp bảo mật thông tin • Nhiệm vụ: <ul style="list-style-type: none"> ○ Tìm hiểu về Model Context Protocol: <ul style="list-style-type: none"> ▪ Nguyên lý hoạt động của MCP ▪ Lợi ích của MCP trong quản lý context ▪ Cách áp dụng MCP với các LLM API ○ Thiết kế cấu trúc context theo MCP: <ul style="list-style-type: none"> ▪ Xác định các thành phần cơ bản: Goals, Context, Guidelines ▪ Tao template cho các phần Facts, Examples, Instructions ▪ Cấu trúc dữ liệu database trong MCP context ○ Triển khai cơ chế bảo mật: <ul style="list-style-type: none"> ▪ Lọc thông tin nhạy cảm trước khi gửi đến LLM ▪ Kiểm soát dữ liệu được truyền trong context ▪ Thiết lập quyền truy cập cho database ○ Thủ nghiệm với nhiều loại truy vấn khác nhau ○ Đánh giá hiệu quả của MCP so với prompt thông thường • Kết quả: <ul style="list-style-type: none"> ○ Template MCP cho AI Agent ○ Cơ chế bảo mật thông tin hoạt động hiệu quả ○ Báo cáo đánh giá hiệu suất khi sử dụng MCP ○ Tài liệu hướng dẫn áp dụng MCP cho các trường hợp khác nhau
4	<p>Giám sát và ghi nhận chi phí API</p> <ul style="list-style-type: none"> • Mục tiêu: Xây dựng hệ thống giám sát chi phí API và tìm hiểu các framework có sẵn • Nhiệm vụ: <ul style="list-style-type: none"> ○ Thiết lập hệ thống ghi nhận sử dụng API: <ul style="list-style-type: none"> ▪ Tạo log ghi lại số lượt gọi API ▪ Ước tính chi phí theo số token sử dụng ▪ Tạo báo cáo đơn giản về tần suất sử dụng ○ Khám phá các framework có sẵn để quản lý chi phí: <ul style="list-style-type: none"> ▪ Tìm hiểu các tùy chọn trong LangChain/Flowise về quản lý token ▪ Nghiên cứu các công cụ monitoring có sẵn của nhà cung cấp LLM ○ Thực hiện các phương pháp đơn giản để kiểm soát chi phí: <ul style="list-style-type: none"> ▪ Giới hạn chiều dài context gửi đến LLM ▪ Sử dụng các mô hình nhỏ hơn cho các tác vụ đơn giản ○ Thiết lập ngưỡng cảnh báo đơn giản khi vượt quá ngân sách • Kết quả: <ul style="list-style-type: none"> ○ Hệ thống ghi nhận cơ bản về sử dụng API ○ Báo cáo về các framework và công cụ giám sát chi phí ○ Báo cáo về chi phí sử dụng và các cách đơn giản để kiểm soát ○ Bảng so sánh chi phí giữa các mô hình LLM khác nhau
5	<p>Phát triển giao diện người dùng và hiển thị record clickable</p> <ul style="list-style-type: none"> • Mục tiêu: Xây dựng giao diện frontend để hiển thị kết quả và record clickable • Nhiệm vụ:

	<ul style="list-style-type: none"> ○ Thiết kế form nhập liệu cho người dùng ○ Phát triển giao diện hiển thị kết quả: <ul style="list-style-type: none"> ▪ Khu vực hiển thị văn bản ▪ Component hiển thị record clickable ▪ Chức năng mở popup hoặc new tab ○ Tích hợp với giao diện xem file đã có sẵn ○ Tạo các template hiển thị cho các loại record khác nhau ○ Thiết kế chức năng filter/sort kết quả <ul style="list-style-type: none"> • Kết quả: <ul style="list-style-type: none"> ○ Giao diện người dùng hoàn chỉnh ○ Component hiển thị record clickable ○ Tích hợp với hệ thống xem file ○ Báo cáo đánh giá trải nghiệm người dùng
6	<p>Tích hợp các thành phần và kiểm thử</p> <ul style="list-style-type: none"> • Mục tiêu: Tích hợp tất cả các thành phần và kiểm thử toàn diện • Nhiệm vụ: <ul style="list-style-type: none"> ○ Kết nối workflow backend với frontend: <ul style="list-style-type: none"> ▪ Thiết lập API endpoint hoặc webhook ▪ Cấu hình xử lý request/response ○ Kiểm thử toàn diện: <ul style="list-style-type: none"> ▪ Kiểm thử độ chính xác của kết quả ▪ Kiểm thử hiệu năng và thời gian phản hồi ▪ Kiểm thử bảo mật và điều khiển quyền truy cập ▪ Kiểm tra tính năng record clickable ○ Xử lý các trường hợp ngoại lệ và lỗi ○ Tối ưu thời gian phản hồi • Kết quả: <ul style="list-style-type: none"> ○ Hệ thống tích hợp hoàn chỉnh ○ Báo cáo kiểm thử chi tiết ○ Danh sách vấn đề và kế hoạch khắc phục ○ Phiên bản beta sẵn sàng cho người dùng thử nghiệm
7	<p>Thu thập phản hồi và cải thiện</p> <ul style="list-style-type: none"> • Mục tiêu: Thu thập phản hồi từ người dùng và cải thiện hệ thống • Nhiệm vụ: <ul style="list-style-type: none"> ○ Tổ chức phiên thử nghiệm với người dùng thực tế ○ Thu thập phản hồi về: <ul style="list-style-type: none"> ▪ Độ chính xác của kết quả ▪ Trải nghiệm sử dụng ▪ Tốc độ phản hồi ▪ Tính hữu ích của record clickable ○ Phân tích dữ liệu sử dụng và pattern tìm kiếm ○ Cải thiện hệ thống dựa trên phản hồi: <ul style="list-style-type: none"> ▪ Tinh chỉnh prompt ▪ Tối ưu caching ▪ Cải thiện giao diện • Kết quả: <ul style="list-style-type: none"> ○ Báo cáo phản hồi từ người dùng ○ Danh sách cải tiến đã thực hiện ○ Phiên bản cải tiến của hệ thống ○ Báo cáo đánh giá hiệu quả sau cải tiến

Hoàn thiện, tài liệu hóa và triển khai

- **Mục tiêu:** Hoàn thiện, tài liệu hóa và chuẩn bị triển khai sản phẩm
- **Nhiệm vụ:**
 - Hoàn thiện những chi tiết cuối cùng:
 - Xử lý các lỗi còn tồn đọng
 - Tối ưu hiệu suất cuối cùng
 - Chuẩn bị môi trường production
 - Tạo tài liệu cho dự án:
 - Tài liệu hướng dẫn sử dụng
 - Tài liệu hướng dẫn quản trị
 - Tài liệu kỹ thuật và cấu hình
 - Chuẩn bị kế hoạch maintenance:
 - Quy trình cập nhật prompt
 - Quy trình quản lý chi phí
 - Kế hoạch backup và restore
 - Chuẩn bị demo và trình bày sản phẩm
- **Kết quả:**
 - Sản phẩm hoàn chỉnh sẵn sàng triển khai
 - Bộ tài liệu đầy đủ
 - Kế hoạch maintenance
 - Bài trình bày demo sản phẩm cuối kỳ

..... (nếu còn)

(Trong phần này, DN có thể xóa bỏ thông tin mẫu bị dù)

II. HỒ SƠ, PHỎNG VẤN, LIÊN HỆ (DN tự nhận và xử lý hồ sơ):

1. Hồ sơ:

- Hồ sơ bao gồm:
 - 1 bản CV giới thiệu về bản thân
 - 1 CCCD bản photo
 - 1 thẻ sinh viên bản photo
 - Địa điểm/ kênh tiếp nhận: Các bạn Sinh viên quan tâm gửi hồ sơ về email: hr@dientoan.vn với nội dung tiêu đề: [TTTN]-[Họ và tên]-[Lớp]
- Hạn nộp: 26/05/2025

2. Phỏng vấn/ xét tuyển: Thời gian phỏng vấn: sẽ được thông báo cụ thể sau khi tổng hợp danh sách đăng ký.

3. Liên hệ:

Bảo Châu (Ms.) – HR

Điện thoại: 0909 935 920

Email: hr@dientoan.vn

III. DN CAM KẾT VỚI KHOA:

(Điều này rất quan trọng, rất mong Quý DN lưu ý hỗ trợ Khoa/ SV)

1. DN sẽ gửi chương trình file scan màu (theo mẫu file D2) có ký tên đóng dấu hoàn chỉnh cho Khoa sau khi chương trình được Khoa duyệt trong vòng 03 ngày.
2. DN sẽ gửi file scan màu (theo mẫu file D3) có ký tên đóng dấu hoàn chỉnh công văn xác nhận SV trúng tuyển thực tập cho Khoa (nếu có SV trúng tuyển) – theo mẫu của Khoa, trong vòng 03 ngày sau khi xác nhận với Khoa/ SV.
3. Hướng dẫn SV làm việc theo đúng chương trình đã thống nhất với Khoa. **Nếu DN có yêu cầu SV trúng tuyển ký bất kỳ cam kết gì khác, thì cần nêu rõ trong phần “Thông tin thêm (nếu có)” trong chương trình (mục I.2).**
4. DN có mentor chuyên môn có trình độ Thạc sĩ, hoặc trình độ Đại học với 5 năm kinh nghiệm trở lên làm đại diện hướng dẫn chương trình thực tập.
5. Sắp xếp tiếp cán bộ giám sát do Khoa cử sang DN vào khoảng giữa kỳ thực tập (nếu có). Cán bộ được Khoa phân công sẽ trực tiếp liên hệ với DN để thống nhất lịch tiếp này. Cán bộ giám sát có trách nhiệm quan sát/ đánh giá chương trình thực tập của DN có đạt hay không so với tiêu chí Khoa đề ra, sau đó kết luận/ đề xuất Khoa có nên duyệt gửi SV đi thực tập tại DN nữa hay không?!
6. Gửi hồ sơ thực tập về cho Khoa trước 16g00 ngày 22/08/2025, theo **mẫu của Khoa** (**bản giấy có ký tên - đóng dấu hoàn chỉnh**), bao gồm:
 - Chương trình thực tập (form D2).
 - Công văn xác nhận kết quả xét tuyển (form D3).
 - Bảng điểm (form D4).
 - Bảng đánh giá SV (form D5).

Địa chỉ nhận hồ sơ bản giấy có ký tên – đóng dấu hoàn chỉnh (đường BUU ĐIỆN):

Khoa KH&KT Máy Tính - Trường Đại học Bách Khoa
268 Lý Thường Kiệt, Phường 14, Quận 10, TPHCM (Nhà A3)
ĐT: 028 3863 8912 - Ext: 7847

Thành phố Hồ Chí Minh, ngày 26 tháng 04 năm 2025

Ban Lãnh đạo



Dương Ngọc Hiếu