A picture containing rectangle, screenshot, text, frame

Description automatically generated

**Môn học: Thực tập doanh nghiệp – IS502.P21**

BÁO CÁO THỰC TẬP

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA HỆ THỐNG THÔNG TIN

**Giảng viên hướng dẫn**: ThS. Hà Lê Hoài Trung

**Công ty thực tập**: Hitachi Digital Services

**Họ và tên sinh viên:** Nguyễn Hoàng Đăng Khoa

**MSSV:** 21520999

TP. HỒ CHÍ MINH, NĂM 2025

**LỜI CẢM ƠN**

Trân trọng gửi lời cảm ơn Công ty TNHH Hitachi Digital Services đã tạo điều kiện cho em có cơ hội được thực tập tại công ty.

Chỉ trong một thời gian ngắn, nhưng nhờ sự chỉ dẫn hỗ trợ của mọi người, em đã tiếp thu được những kiến thức quan trọng để có thể trực tiếp tham gia vào dự án, nơi em có cơ hội làm việc cùng với các thành viên khác. Qua việc tham gia vào dự án, em đã hiểu rõ hơn về quy trình phát triển phần mềm, quy trình làm việc, xử lý logic nghiệp vụ cho đến triển khai ứng dụng trên môi trường thực tế.

Em cũng xin gửi lời tới anh Trần Văn Phi (CTO - Chief Technology Officer) giám đốc công nghệ của chi nhánh Hitachi Digital Services tại thành phố Hồ Chí Minh, đã trạo cho cơ hội tham gia dự án thực tế của công ty. Và đặc biệt cảm ơn anh Bảo Trần (Bao Tran– Team lead), anh Lê Quốc Anh (David Le – Sub team leader), chị Huỳnh Thanh Thảo (Huynh Thanh Thao – Sub team leader) và mọi người trong dự án đã hướng dẫn, giúp đỡ em tận tình cả những khó khăn trong công việc cũng như những khó khăn việc làm quen với môi trường mới. Nhờ sự hướng dẫn và giúp đỡ của mọi người, em đã có thể hoàn thành nhiệm vụ được giao một cách hiệu quả và đạt được mục tiêu của dự án.

Cũng như cảm ơn thầy cô trong khoa Hệ thống thông tin đã nhiệt tình hỗ trợ, tạo điều kiện cho em làm bài báo cáo này.

Em rất biết ơn những cơ hội quý giá mà Công ty TNHH Hitachi Digital Services và khoa Hệ thống thông tin đã mang đến cho em. Nhờ những trải nghiệm này, em đã không chỉ nắm bắt được những kiến thức chuyên môn mà còn phát triển kỹ năng làm việc nhóm, giao tiếp và giải quyết vấn đề. Điều này sẽ góp phần quan trọng trong sự nghiệp và phát triển tương lai của em.

Một lần nữa, em xin gửi lời cảm ơn chân thành đến Công ty TNHH Hitachi Digital Services, các thành viên trong dự án và toàn thể thầy cô trong khoa Hệ thống thông tin. Em sẽ luôn ghi nhớ những kinh nghiệm và bài học quý báu mà em đã học được từ thời gian thực tập này.

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

Họ và tên sinh viên: Nguyễn Hoàng Dăng Khoa

MSSV: 21520999

**Đánh giá quá trình thực tập của sinh viên:**

Danh mục

[Chương 1: Giới thiệu 1](#_Toc200989928)

[I. Giới thiệu công ty 1](#_Toc200989929)

[II. Dự án, trách nhiệm công việc 2](#_Toc200989931)

[a. Vị trí thực tập 2](#_Toc200989932)

[b. Lịch và địa điểm công tác thực tập 3](#_Toc200989933)

[Chương 2: Quá trình thực tập 4](#_Toc200989934)

[I. Mốc thời gian giao việc và kết quả 4](#_Toc200989935)

[Chương 3: Cơ sở lý thuyết – Công nghệ sử dụng 21](#_Toc200989936)

[I. Python 21](#_Toc200989937)

[II. Amazon Web Service 22](#_Toc200989938)

[a. AWS API Gateway 23](#_Toc200989939)

[b. AWS SAM 24](#_Toc200989940)

[c. AWS Lambda Function 25](#_Toc200989941)

[Chương 4: Thông tin chi tiết công việc 27](#_Toc200989942)

[I. Thông tin dự án 27](#_Toc200989943)

[**a.** Thông tin chi tiết dự án: **Revoria Cloud Production GenAI** 27](#_Toc200989944)

[b. Nhóm: 28](#_Toc200989945)

[c. Phân công công việc: 28](#_Toc200989946)

[II. Quá trình công việc thực hiện 29](#_Toc200989947)

[a. Giai đoạn training: 29](#_Toc200989948)

[b. Công việc thực hiện: 30](#_Toc200989949)

[Chương 5: Tổng kết 37](#_Toc200989950)

[I. Tổng kết: 37](#_Toc200989951)

[II. Thuận lợi: 37](#_Toc200989952)

[III. Khó khăn gặp phải: 37](#_Toc200989953)

[IV. Đóng góp cho công ty: 37](#_Toc200989954)

[Tài liệu tham khảo 39](#_Toc200989955)

Danh mục hình ảnh

[Hình 1. Logo công ty Hitachi 1](#_Toc200733436)

[Hình 2. Thông tin thực tập 2](#_Toc200733437)

[Hình 3. Python 15](#_Toc200733438)

[Hình 4. Amazon Web Services 16](#_Toc200733439)

[Hình 5. AWS API Gateway 16](#_Toc200733440)

[Hình 6. AWS SAM 18](#_Toc200733441)

[Hình 7. AWS Lambda function 18](#_Toc200733442)

[Hình 8. Thông tin dự án 20](#_Toc200733443)

[Hình 9. Sơ đồ phân cấp trong team dự án 21](#_Toc200733444)

[Hình 10. Phân công công việc 21](#_Toc200733445)

[Hình 11. Màn hình 20 22](#_Toc200733446)

[Hình 12.Màn hình 21 22](#_Toc200733447)

[Hình 13. Dataflow dự án 23](#_Toc200733448)

[Hình 14. GitLab dự án 24](#_Toc200733449)

[Hình 15. Hàm refDataSearch phiên bản đầu tiên 24](#_Toc200733450)

[Hình 16. Tích hợp với lambda và thay đổi theo thiết kế 25](#_Toc200733451)

[Hình 17. Xây dựng API số 1 thông qua Boto3 26](#_Toc200733452)

[Hình 18. Xây dựng thông qua CloudFormation tích hợp AWS SAM 27](#_Toc200733453)

[Hình 19. Xây dựng logic Step function 28](#_Toc200733454)

[Hình 20. Xây dựng API số 3 cho Step Function 29](#_Toc200733455)

[Hình 21. Kiểm thử API số 2 29](#_Toc200733456)

[Hình 22. Kiểm thử kết quả trả về Step Function 30](#_Toc200733457)

[Hình 23. Commit trên nhánh feature/basic-information-conversion 30](#_Toc200733458)

[Hình 24. Commit trên nhánh feature/similar-job-searching 31](#_Toc200733459)

[Hình 25. Commit trên nhánh feature/data-filling 31](#_Toc200733460)

# Chương 1: Giới thiệu

## Giới thiệu công ty



Hình . Logo công ty Hitachi

Website công ty: <https://www.hitachids.com/careers/our-people/vietnam/>

Công ty khởi nguồn tại California (Mỹ) với tên Global CyberSoft, chuyên phát triển phần mềm nhúng, ứng dụng doanh nghiệp và kiểm thử phần mềm. Về sau Hitachi Consulting (thuộc Tập đoàn Hitachi) mua lại Global CyberSoft, Việt Nam gia nhập hệ sinh thái toàn cầu. Trong giai đoạn 2014–2016: công ty thành lập chi nhánh tại Đà Nẵng (2014) và Hà Nội (2016); quy mô lên tới 1.000 nhân viên vào năm 2015.

Mục tiêu tập trung phát triển bộ phận “Applied New Technology”, hợp tác cùng SAP, Microsoft, AWS, etc... nhằm cung cấp giải pháp chuyển đổi số và tự động hóa cho doanh nghiệp.

Vào tháng 1/2020, Hitachi Consulting hợp nhất với Hitachi Vantara toàn cầu, đổi tên tại Việt Nam thành Hitachi Vantara Vietnam. Sau đó, đổi tên tiếp thành Hitachi Digital Services Vietnam, định vị là trung tâm chuyển đổi số mạnh hỗ trợ từ kỹ thuật số, dữ liệu đến cơ sở hạ tầng, bao gồm cả tích hợp SAP S/4HANA, hệ thống nhúng, kiểm thử tự động, AI/ML…

HitachiDSVN cung cấp đa dạng dịch vụ và giải pháp công nghệ, bao gồm:

* **Dịch vụ đám mây và dữ liệu**: Hỗ trợ doanh nghiệp trong việc di chuyển, quản lý và tối ưu hóa hạ tầng đám mây và dữ liệu.
* **Dịch vụ IoT**: Triển khai các giải pháp Internet vạn vật để nâng cao hiệu quả hoạt động và kết nối thiết bị.
* **Ứng dụng doanh nghiệp số**: Phát triển và tích hợp các ứng dụng nhằm cải thiện quy trình kinh doanh.
* **Dịch vụ quản lý**: Cung cấp dịch vụ quản lý hệ thống CNTT toàn diện, giúp doanh nghiệp tập trung vào hoạt động cốt lõi.
* **Dịch vụ nhúng**: Phát triển phần mềm nhúng cho các thiết bị và hệ thống chuyên dụng.
* **Trung tâm độ tin cậy ứng dụng (HARC)**: Đảm bảo độ tin cậy và hiệu suất của các ứng dụng doanh nghiệp.



## Dự án, trách nhiệm công việc

### Vị trí thực tập

* Vị trí thực tập: Data Engineer Intern.
* Trách nhiệm công việc: Hỗ trợ triển khai RestAPI, Lambda function bằng ngôn ngữ Python, cấu hình pipeline đảm bảo dữ liệu theo sát thiết kế Data flow. Kiểm thử hệ thống trên cơ sở hạ tầng đám mây Amazon Web Services (AWS) với thư viện Python SDK Boto3.

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Thông tin thực tập

### Lịch và địa điểm công tác thực tập

* Địa điểm làm việc: Tầng 9, tòa Orbital, Khu Công viên phần mềm Quang Trung, P. Tân Chánh Hiệp, Quận 12, TP.HCM
* Hình thức làm việc: Part time.
* Thời gian thực tập: 15/04/2025 – 15/07/2025 (3 tháng).
* Thời gian làm việc: Thứ 2, thứ 4, thứ 5, thứ 6.
  + Buổi sáng: 9 giờ – 12 giờ.
  + Buổi chiều: 13 giờ - 18 giờ.

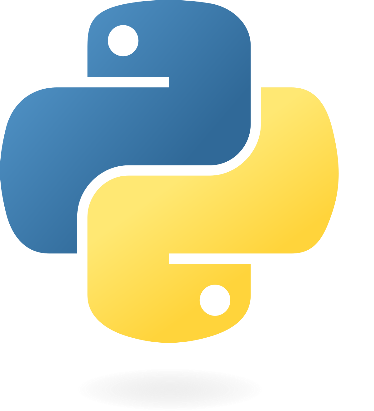
# Chương 2: Quá trình thực tập

## Mốc thời gian giao việc và kết quả

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Timeline | Assigned task(s) | Achieved result(s) | Experience | Further research |
| 15/04/2025 | **Quá trình Onboarding** | | | |
|  | Set up máy tính |  |  |  |
| 16/04/2025 | **Generative Ai tự học** | | | |
|  | * Cài flowise. * Tạo flowchat trong flowise. * Viết test case CV generation cho HR (input và expected output) | * Hoàn thành viết test case. * Flowise đã cài đặt thành công globally nhưng khi bấm icon chat để thử thì bị lỗi UI. |  |  |
| 17/04/2025 |  |  |  |  |
|  | * Sửa lại flowise. * Tạo flowchat trong flowise. * Cài đặt Ollama và test thử API. | * Sử dụng được Flowise local host, không còn bị lỗi UI. * Olama cài thành công và API test thành công. | * Phải downgrade node js từ v22 xuống v20 để install flowise thành công. * LLM OpenAI không sử dụng được trong flowchat do policy network công ty, Ollama thì dùng dược. * Parameter Temperature từ 0 đến 1 dùng để biểu thị cho tính creative của model (Ví dụ 0 thì sẽ very factual còn 1 thì sẽ creative) | * Tìm cách sử dụn OPEN AI LLM Api key |
| 18/04/2025 |  |  |  |  |
|  | * Summarize các loại Prompt. * Tìm cách sử dụng OPEN AI LLM nếu được. | * Hoàn thành summarize các loại prompt. * Không sử dụng được OPEN AI LLM, phải sử dụng Ollama. |  |  |
| 21/04/2025 |  |  |  |  |
|  | * Tìm cách làm 1 chat bot summarize document. | * Thành công tạo được chat bot summarize document. * Tuy nhiên summarize document size nhỏ (vd: CV, PDF) hay các file chứa nhiều hình ảnh trả về kết quả không chính xác bị hallucination cao, còn document size lớn hơn (vs: report 50 trang) thì summarize được. | * Chat flow cần vector store để chưa chứa document được embedding thành vector. | * Tìm cách sử dụng các vector store. * Tìm cách làm chat bot có long term history. |
| 22/04/2025 | **Cài đặt, kiểm thử và báo cáo AI Coder Hỗ trợ TabbyML trên môi trường Vscode.** | | | |
|  | * Thử install tabby trên VS Code để test các feature như chat vs nó, code completion. * Tham gia orientation training của công ty P1. * Cải tiến chat bot có thể summarize được file pdf lớn hơn và implement thêm chat history cho chat bot. | * Các Feature của Tabby hoạt động thành công trên VSCode nhưng quá trình chạy lâu. * Nắm được cấp bậc trong cty, các CoE, các bộ phận trong cty, các quy trình workflow. * Chat Bot đã có thể summarize file report lớn. Chat history đã hoạt động, tuy nhiên chỉ lưu trong memory nên khi khởi động lại flowise thì mất hết. | * Sử dụng Pinecone để làm để làm vector database cho chat flow. Pinecone hỗ trợ tải lên file trong cửa sổ chat. * Buffer Memory có thể giúp lưu lại chat history trong 1 thơi gian trước khi flowise được tắt đi và khởi động lại. | * Có gắng tải lên được file lên vector database. * Thử chat memory khác cho chat flow để có thể lưu được lâu dài. |
| 23/04/2025 |  |  |  |  |
|  | * Tham gia orientation trainging của công ty P2. * Cải tiến chat bot có thể upload file lên vector database và cải thiện chat thông tin trí nhớ. | * Nắm được các chính sách của công ty về bảo hiểm, quyền lợi như gửi xe, ngày nghỉ, etc... * Đã có thể upload lên cloud của pinecone tuy nhiên chưa kéo được dữ liệu về. Container Redis database chạy trên local có thể lưu được lịch sử chat tưu nhiên nếu như đẩy file pdf lớn thì Redis sẽ lưu tất cả vào 1 index gây ra vấn đề overflow, dẫn đến vấn dề Hallucination hay model không nhớ. |  | * Cố gắng cải thiện chat flow để có rag được dữ liệu. * Thử sử dụng cách summarize file trước khi bỏ vào redis chat memory. |
| 24/04/2025 | **Cài đặt và viết hướng dẫn cài đặt TabbyML trên môi trường IntelliJ** | | | |
|  | * Thử install tabby trên Intellij, chạy tabby trên local, và viết báo cáo hướng dẫn install tabby trên Intellij. * Cải tiến chat bot để thể rag dữ liệu từ vector database. | * Sử dụng Tabby thành công trên Intellij và tốc độ respond vẫn chậm như sd trên VS Code. Báo cáo hương dẫn và Demo của Tabby được chấp nhận bởi thành viên IT hỗ trợ. * Thành công cải tiến chat bot mang dữ liệu từ redis stack database chạy trên local. |  | * Cố gắng nâng cấp dữ bộ nhớ để có thể sumamrize file trước khi bỏ vào redis chat memory. |
| 28/04/2025 |  |  |  |  |
|  | * Đưa ra 1 proposal về 1 vấn đề của internal project đề ra: khi có lượng lớn training video, có thể sử dụng GenAI để có thể tìm kiếm video và timestamp. | * Đưa ra được proposal về việc transcribe được phụ đề có timestamps kèm theo thông tin qua sử dụng mô hình dữ liệu lớn. * Đưa ra được kiến trúc hệ thống để giải quết vấn đến. |  |  |
| 29/04/2025 |  |  |  |  |
|  | * Dùng bất cứ tool hay công nghệ nào có thể extract được content từ 1 audio | * Sử dụng ffmpeg để có thể triết xuất âm thanh từ video. * Sử dụng và so sánh giữa hai mô hình PhoWhisper và OpenWhisper để để có thể triết xuất file âm thanh thành dạng chữ bằng ngôn ngữ Python. | * Cần phải có video rõ ràng thì mới extract được transcript (không thể triết xuất được transcript từ video có âm thanh không rõ ràng). | * Lựa chọn vector database để có thể embedding dữ liệu từ hai model |
| 30/4 – 1/5 | **Nghỉ lễ 30/4 – 1/5** | | | |
| 5/5/2025 | **Tham gia project FlexBA để hỗ trợ quản lý TabbyML** | | | |
|  | * Tham gia project FlexBA để hỗ trợ quản lý TabbyML server. |  | * Sử dụng SSH để liên kết tới server host TabbyML. | * TabbyML có bug dẫn đến database bị khóa. Cần phải tìm lý do dẫn đến bug (root cause) |
| 6/5/2025 | **Tham gia project Revoria Cloud Production GenAI** | | | |
|  | * Sử dụng 3 cách tiếp cận Similarity Search: Fuzzywuzzy + Levenshtein, Nano-vectorDB, PandasAI với ngôn ngữ Python và đánh giá kết quả đầu ra. | * PandasAI không sử dụng để similar search được. * Chưa embed dữ liệu dc data cho Nano-vectorDB. | * Khó khăn trong việc embedding khi dữ liệu đầu vào là tiếng Nhật (input\_search\_DB.csv) | * Tìm kiếm phương án embed dữ liệu, đặc biệt khi dữ liệu đầu vào là chữ Nhật. |
| 7/5/2025 |  |  |  |  |
|  | * Tìm phương án embed data và đẩy dữ liệu lên Nano-vectorDB. | * Embed dữ liệu thành công bằng cách sử dụng model ollama nomic-embed-text và multilingual-e4-small. * Upsert lên Nano-vector và query thành công vs thời gian lần lượt là 0.0015s và 0.0004s cho in 100 dòng. |  |  |
| 8/5/2025 |  |  |  |  |
|  | * Đánh giá thời gian đẩy và truy vấn data trên Nano-vectorDB với dữ liệu ~1 triệu dòng. | * Do là sử dụng model nomic-embed-text nên ra đầu ra chiều của vector quá lớn (768 dims) => gắn dữ liệu vào list tốn nhiều thời gian (2h45m), còn model multilingual-e4-small (384 dims) nhanh hơn do có chiều vector nhỏ hơn 15s | * Nếu mỗi bộ dữ liệu nhỏ (Dạng câu, không phải dạng file lớn) nhưng lượng dữ liệu lớn thì có thể giảm chiều dữ liệu vector để có thể giảm thời gian xử lý. |  |
| 12/5/2025 |  |  |  |  |
|  | * Tìm hiểu cấu trúc, thiết kế dataflow, thiết kế database, danh sách RestAPI. * Tìm hiểu về cách dịch vụ AWS sẽ được sử dụng trong project (AWS API Gateway, AWS S3, AWS Aurora, AWS Lambda, AWS Bedrock) * Tìm hiểu về thư viện python RAGAS, công cụ dùng để đánh giá mô hình LLM. |  |  | * Được thêm vào GitLab của dự án. Đọc và tìm hiểu soure code python của dự án. |
| 15/5/2025 |  |  |  |  |
|  | * Thuyết trình về các dịch vụ AWS sẽ được sử dụng trong đồ án cho người hướng dẫn. * Đọc và tìm hiểu source code python của đồ án. | * Hiểu được cách hoạt động, cách tính tiền của các phương án của các dịch vụ. |  | * Tìm hiểu cách các dịch vụ AWS được tích hợp vào dự án. |
| 19/5/2025 |  |  |  |  |
|  | * Họp cùng thành viên trong nhóm với Team lead và senior support. Khái quát cơ bản data flow dự án, logic xử lý dữ liệu. * Xây dựng hàm SimilarSearching theo phương án khoảng cách Levenshtein theo trên máy local. | * Xây dựng khung xương của function nhưng kết quả kết quả dầu ra bị lỗi. * Gặp khó khăn trong xử lý tag file XML bằng thư viện Etree của python | * Khó khăn trong việc xây dựng logic xử lý dữ liệu do source code được chú thích bằng tiếng Nhật. * Không nắm rõ thiết kế front end do rào cản ngôn ngữ. * Thiếu kinh nghiệm trong việc xử lý file XML bằng Python | * Tìm hiểu sâu thư viện Etree của Python. * Trao đổi lại thành viên trong nhóm để thống nhất thiết kế logic. |
| 20/5/2025 |  |  |  |  |
|  | * Tiếp tục hoàn thiện hàm SimilarSearching trên local máy. * Thay đổi dữ liệu đầu vào và dữ liệu đầu ra cho chuẩn bị bước tiếp theo chuyển thành AWS Lambda function | * Hoàn thành hàm SimilarSearch. Kiểm thử kết quả đúng với đầu ra của data flow dự án. |  |  |
| 21/5/2025 | * Senior Support – Anh David Le cung cấp AWS Access Key và AWS Secret Access Key. * Xây dựng hàm Lambda SimilaSearchAWS để kiểm thử trên hạ tầng cloud AWS. | * Xây dựng được hàm Lambda gặp lỗi trong quá trình kiểm thử hoạt động trên hạ tầng cloud AWS. | * Khó khăn khi làm việc với AWS Command Line (AWS CLI) và thư viện Boto3 của Python mà không có dashboard để theo dõi. * Thiếu kinh nghiệm với việc sử dụng AWS Lambda function. | * Tìm hiểu sâu về AWS Lambda function. * Tìm ra lỗi gây ra và khắc phục. |
| 23/5/2025 |  |  |  |  |
|  | * Được cung cấp dashboard AWS để tìm ra lỗi trong Lambda function. * Tìm phương án xử lý. * Bắt đầu xây dựng RestAPI thông qua AWS API Gateway bằng Python, kết nối tới các dịch vụ AWS khác như S3. | * Lỗi do AWS Lambda function sử dụng layer như môi trường để hoạt động (phải thêm layer Python 3.12) |  |  |
| 26/5/2025 |  |  |  |  |
|  | * Tiếp tục xây dựng API để có thể gọi tới hàm Lambda SimilarSearchAWS (API 2). | * API triển khai thành công nhưng kiểm thử lại trả ra lỗi mặc dù đã gọi được hàm Lambda | * Thiếu kinh nghiệm làm việc với thư viện Boto3 để tương tác với dịch vụ AWS thông qua Python. | * Tìm ra lỗi và phương hướng giải quyết. |
| 28/5/2025 |  |  |  |  |
|  | * Tìm ra lỗi và cách khắc phụ cho API 2. * Bắt đầu triển khải API cho hàm Lambda BasicInformationConversion (API 1) của thành viên trong nhóm. | * Khắc phục tạm thời bằng cách chuyển từ AWS\_PROXY sang AWS (non proxy). Cần phải tự mapping đầu vào và đầu ra của API khớp với Lambda. |  |  |
| 29/5/2025 |  |  |  |  |
|  | * Hoàn thiện API 1 và API 2 như trong thiết kế của dự án. * Demo API 1 và API 2 cho Team Lead Bao Tran, đẩy code lên repository dự án trên GitLab. | * Thành công triển khai API 1 và API 2 trên hạ tầng cloud. |  |  |
| 30/5/2025 |  |  |  |  |
|  | * Xây dựng API để gọi tới AWS Step Function. * Viết file yaml CloudFormation chuẩn bị triển khai trên AWS SAM để kiểm thử trên môi trường local. | * API gọi tới Step Function thành công nhưng chưa trả về được kết quả cho UI bên team Front End. * File yaml build gặp lỗi, hàm Lambda và API không hoạt động trong môi trường local. | * Thiếu kinh nghiệm trong việc sử dụng CloudFormation yaml dẫn đến triển khai trên AWS SAM lỗi. |  |
| 2/6/2025 |  |  |  |  |
|  | * Kiểm tra lỗi và cách khắc phục lỗi trong CloudFormation trên AWS SAM. | * Lỗi do lệnh sam local start-api không thể xử lý dữ liệu đầu vào như chuẩn json. * Đổi lại từ AWS (non proxy) thành AWS\_PROXY. Thay đổi xử lý bắt sự kiện của hàm Lambda cho phù hợp. | * Hiểu được giao tiếp giữa API Gateway và hàm Lambda khi sử dựng RestAPI. * Hiểu được lỗi lúc ban đầu triển khai API Gateway gọi tới hàm Lambda |  |
| 4/6/2025 |  |  |  |  |
|  | * Thay đổi lại dữ liệu đầu vào và đầu ra của API số 1 và API số 2 do team Front End đề nghị theo yêu cầu của khách. * Tìm hiểu dể triển khai AWS Cognito cho API Gateway. * Tích hợp sử dụng PostregSQL để chuẩn bị triển khai AWS Aurora. | * Sửa đổi API 2 dữ liệu đầu ra, they đổi logic xử liệu trả về. Chưa thể kiểm thử do thiếu dummy data. |  |  |
| 5/6/2025 |  |  |  |  |
|  | * Thay đổi API URL theo đúng thiết kế. * Format lại code của hàm lambda, sử dụng packages utils trả về messages và lỗi của team dev. | * Kiểm thử trên máy local thông qua sam không thành công. Lỗi python không import được các package hỗ trợ dù không báo lỗi trên IDE. |  |  |
| 9/6/2025 |  |  |  |  |
|  | * Sửa lỗi sam local start-api chạy với packages utils. * Triển khai API số 3 với Step function cho bước cuối Data Filling | * Sữa lỗi thành công, đẩy các package uilts chung với lambda function. * API số 3 triển khai thành công, khởi động được AWS Step function nhưng không thể trả về kết quả theo thiết kế. | * Thiếu kinh nghiệm trong việc sử dụng chuyên sau các dịch vụ AWS dẫn đến liên kết gặp khó khăn khi ráp nhiều dịch vụ lại với nhau (Lambda Function, API Gateway, Step Function) | * Tìm ra nguồn lỗi của trả về không đúng của API khi kết hợp với Step Function. |
| 11/06/2025 |  |  |  |  |
|  | * Tìm ra nguyên nhân dẫn đến lỗi của khi API số 3 tích hợp với Step function. | * Tìm ra được nguyên nhân Step function không trả về đúng kế quả mong muốn. | * AWS API Gateway có timeout cao nhất là 29 giây (có thể nâng lên nhưng phải liên hệ AWS), Step function chạy lâu hơn 30s dẫn đến lỗi. * API kích hoạt Step function sử dụng StartExecution chỉ trả về executionArn và startDate (theo hướng dẫn AWS developer guide). |  |
| 13/06/2025 |  |  |  |  |
|  | * Đưa ra hướng giải quyết tạm thời cho API số 3. * Tích hợp API số 1 và số 2 vào swagger sử dụng OpenAPI để chuẩn bị cho Release | * Tạm thời giải quyết thông qua tạo thêm một method GET cho API sử dụng DescribeExecution để gọi tới StepFunction khi hoàn thành quá trình. |  |  |
| 16/6/2025 |  |  |  |  |
|  | * Tìm cách giải quyết |  |  |  |

# Chương 3: Cơ sở lý thuyết – Công nghệ sử dụng

## Python



Hình . Python

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

* Cú pháp đơn giản và dễ đọc, phù hợp cho người mới học lập trình và cũng dễ dàng cho những người có kinh nghiệm.
* Python được sử dụng trong nhiều lĩnh vực như phát triển web, phân tích dữ liệu, trí tuệ nhân tạo, khoa học dữ liệu và nhiều ứng dụng khác. Điều này làm cho Python trở thành một ngôn ngữ lập trình linh hoạt và tiện lợi.
* Cộng đồng Python khá lớn và luôn hoạt động tích cực nhằm cung cấp nhiều thư viện, framework hữu ích. Người dùng sẽ được nhận sự hỗ trợ từ cộng đồng thông qua tài liệu, diễn đàn và các nguồn thông tin trực tuyến.
* Python cung cấp nhiều thư viện mạnh mẽ cho phân tích dữ liệu như Pandas, NumPy, và Matplotlib. Bộ ngôn ngữ giúp người dùng xử lý và thể hiện dữ liệu một cách hiệu quả.
* Python có thể chạy trên nhiều hệ điều hành khác nhau và trở thành một trong những ngôn ngữ lập trình chéo phổ biến.
* Ngôn ngữ Python là mã nguồn mở và miễn phí, cho phép người dùng tự do sử dụng, phân phối và thay đổi phiên bản của nó.

## Amazon Web Service

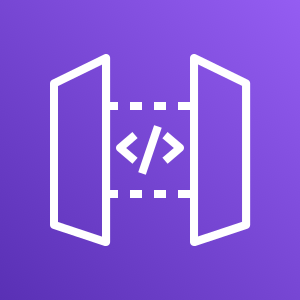


Hình . Amazon Web Services

Amazon Web Services (AWS) là một nền tảng dịch vụ điện toán đám mây (cloud computing) do Amazon phát triển và cung cấp. AWS cung cấp một loạt các dịch vụ theo mô hình “pay-as-you-go” – tức là chỉ phải trả tiền cho những gì sử dụng. Đây là một trong những nền tảng điện toán đám mây phổ biến và lớn nhất trên thế giới.

* Lợi ích của việc sử dụng AWS:
* Tiết kiệm chi phí: Không cần đầu tư máy chủ vật lý.
* Mở rộng linh hoạt: Có thể tăng giảm tài nguyên theo nhu cầu.
* Toàn cầu hóa nhanh chóng: AWS có nhiều trung tâm dữ liệu trên khắp thế giới.
* Bảo mật cao: Đạt nhiều chứng nhận bảo mật quốc tế.

### AWS API Gateway



Hình . AWS API Gateway

AWS API Gateway là một dịch vụ được quản lý hoàn toàn của Amazon Web Services, cho phép bạn tạo, công bố, duy trì, giám sát và bảo mật các API ở bất kỳ quy mô nào. Nó là “cổng” trung gian giữa người dùng (hoặc ứng dụng) và dịch vụ backend như Lambda, EC2, hoặc bất kỳ endpoint HTTP nào.

Nhiệm vụ chính API Gateway là:

* Tạo REST API, HTTP API và WebSocket API
  + REST API: Thích hợp cho các ứng dụng web và mobile.
  + HTTP API: Nhẹ hơn, nhanh hơn và rẻ hơn REST API (hạn chế tính năng).
  + WebSocket API: Hỗ trợ kết nối thời gian thực (real-time), như chat, game,...
* Tích hợp với các dịch vụ backend
  + AWS Lambda: Serverless function.
  + EC2 / ECS / Fargate: Máy chủ hoặc container.
  + HTTP endpoint: Các API hoặc website bên ngoài.
* Quản lý bảo mật
  + API Key, IAM roles, Cognito User Pools, JWT,...
  + Hỗ trợ throttle (giới hạn số request), quota,...
* Chuyển đổi dữ liệu đầu vào/đầu ra: Có thể mapping, chuyển đổi định dạng (JSON → XML,...), hoặc chèn header,...
* Giám sát và logging: Kết hợp với CloudWatch để theo dõi lỗi, thời gian phản hồi,...

### AWS SAM



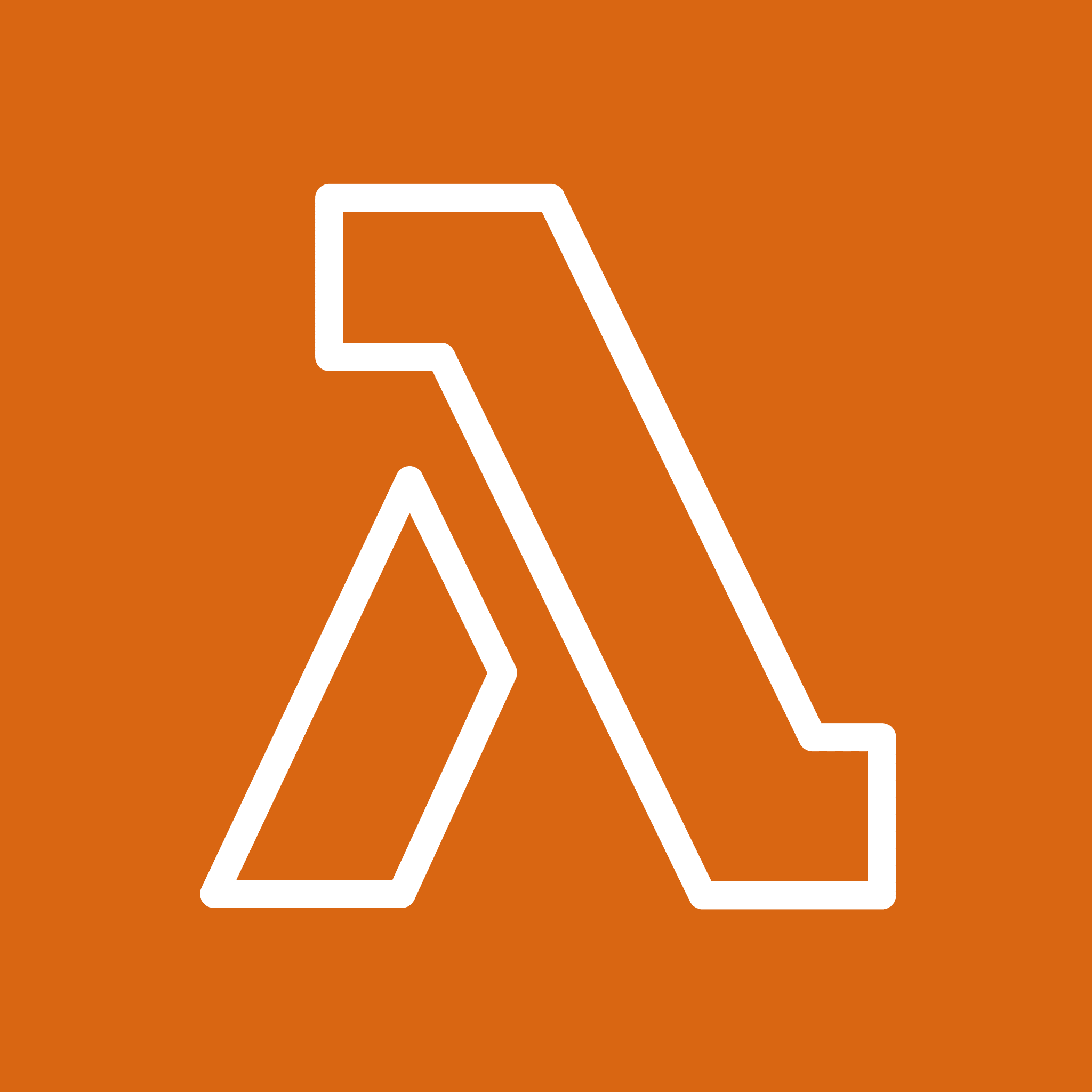
Hình . AWS SAM

AWS SAM (Serverless Application Model) là một framework mã nguồn mở của Amazon Web Services giúp xây dựng, kiểm thử và triển khai ứng dụng serverless dễ dàng hơn. Nó cung cấp một cú pháp ngắn gọn để định nghĩa các tài nguyên như AWS Lambda, API Gateway, DynamoDB, S3... thông qua file cấu hình YAML.

Chức năng chính:

* template.yaml: File cấu hình (theo cú pháp SAM), định nghĩa tài nguyên AWS.
* sam build: Biên dịch mã nguồn, xử lý dependency (nhất là với Python, Node.js).
* sam local: Chạy và kiểm thử Lambda/API Gateway trên máy cục bộ.
* sam deploy: Triển khai ứng dụng lên AWS CloudFormation.

### AWS Lambda Function



Hình . AWS Lambda function

AWS Lambda là một dịch vụ tính toán serverless (không máy chủ) của Amazon Web Services, cho phép chạy mã mà không cần quản lý máy chủ. Chỉ cần viết hàm (function), upload lên AWS, và Lambda sẽ tự động:

* Chạy mã khi có sự kiện (event) xảy ra.
* Tự động cấp phát tài nguyên (RAM, CPU).
* Tự mở rộng quy mô (scale).
* Chỉ trả tiền cho thời gian chạy thực tế của hàm (theo mili giây).

Cấu trúc cơ bản của hàm lambda:

|  |
| --- |
| def lambda\_handler(event, context):  name = event.get("name", "world")  return {  "statusCode": 200,  "body": f"Hello, {name}!"  } |

* event: Dữ liệu đầu vào (do API Gateway, S3, DynamoDB,... gửi tới).
* context: Thông tin về runtime và môi trường thực thi.

Lambda được kích hoạt (trigger) bởi một số sự kiện như:

* API Gateway: Khi có HTTP request (REST API, WebSocket API)
* S3: Khi có file được upload, xóa,...
* DynamoDB Streams: Khi có dữ liệu mới ghi vào bảng
* CloudWatch Events: Lên lịch (cron job, theo giờ/ngày/tuần)
* SNS / SQS: Khi có tin nhắn được gửi
* Step Functions: Kết hợp nhiều hàm Lambda thành quy trình.

# Chương 4: Thông tin chi tiết công việc

## Thông tin dự án

### Thông tin chi tiết dự án: **Revoria Cloud Production GenAI**

A screen shot of a computer

AI-generated content may be incorrect.

Hình . Thông tin dự án

### Nhóm:

A diagram of a company

AI-generated content may be incorrect.

Hình . Sơ đồ phân cấp trong team dự án

### Phân công công việc:

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Phân công công việc

## Quá trình công việc thực hiện

### Giai đoạn training:

* Thời gian ước tính: 1 tuần.
* Tổng quan:
  + Để có thể bắt đầu hỗ trợ triển khai các components trên cơ sở hạ tầng cloud, and David Le đã gửi các tài liệu liên quan đến AWS như developer guide to create RestAPI, AuroraDB, Step Function. Anh Bao Tran cũng yêu cầu tìm hiểu và tạo một buổi thuyết trình qua Microsoft Team để trình bày cho anh chị hiểu về những gì nhóm học được về các dịch vụ AWS trong dự án.

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Slide tìm hiểu về API Gateway và AURORADB

* + Ngoài ra nhóm intern cũng được thêm vào gitlab dự án công ty để có thể hiệu được yêu cầu business logic của khách hàng thông qua code prototype gửi cho công ty.

A screenshot of a computer

AI-generated content may be incorrect.

Hình . GitLab dự án

### Công việc thực hiện:

#### Xây dựng lambda function của function refDataSearch:

* Nhiệm vụ chính: Xây dụng được logic tìm kiếm job gần giống với target\_job sử dụng thuật toán Levenshtein. Trong quá trình xây dựng hàm tìm kiếm cần trao đổi giữa các anh chị liên tục do có thay đổi trong scope dự án theo yêu cầu của bên phía khách hàng.
* Việc xây dựng cần phải tương tác với các dữ liệu file XML mẫu từ phía khách hàng, bắt các thông tin thông qua nhãn dán (tag) XML bằng cách sử dụng thư viện etree của Python. Các dữ liệu XML mẫu có thể bị cũ (outdated) dẫn dến tương tác file đôi lúc bị lỗi, không có thông tin hay tag XML không tồn tại.
* Tính toán điểm gần giống (similar score) sử thụng thư viện module Levenshtein của thư viện textdistance.

A screenshot of a chat

AI-generated content may be incorrect.

Hình . Trao đổi với anh David Le về input và output API số 2

#### Xây dựng API Gateway để gọi hàm Lambda:

* Nhiệm vụ chính: Liên kết API Gateway để invoke hàm Lambda.
* Kiểm thử triển khai trên AWS thông qua thư viện Boto3, sau đó sử dụng CloudFormation bằng cách viết file yaml triển khai các resource cần thiết để phục vụ cho API dựa vào AWS SAM
* Configure đầu vào và đầu ra của API để tương ứng kết sự kiện (event) bắt được của bên phía hàm lambda. => Tường lửa công ty cũng chặn các liên kết bên ngoài mà chưa có được chứng thực bởi ISMS, dẫn đến khó khăn gọi API lỗi.

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Trao đổi chị Thao về kiểm thử API số 1

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Nhánh basic-information-conversion

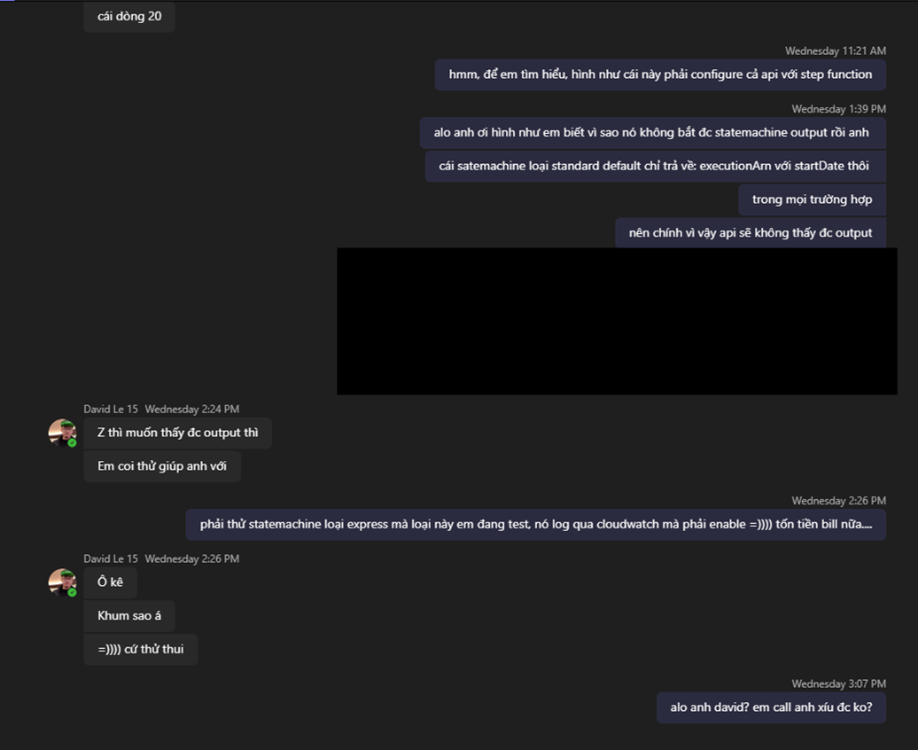
A screenshot of a computer

AI-generated content may be incorrect.

Hình . Kiểm thử API số 2

#### Xây dựng Step function + API cho hàm lambda DataFilling:

* Nhiệm vụ chính: Kết hợp cả 3 dịch vụ AWS (AWS Lambda, AWS API Gateway, AWS Step Function), do mỗi thành viên được giao từng dịch vụ dẫn đến tích hợp khó khăn trong việc triển khai kết hợp, đặt biệt giới hạn của Step function về kết quả trả về, thời gian chạy lâu hơn thời gian sống của API Gateway.
* Statemachine của Step function không thể bị gián đoạn giữa chừng nhưng API cần phải trả về kết quả hàm lambda phía trong Step function cho bên phía UI trong khi phần còn lại chạy ngầm.
  + Vấn đề phát sinh: Action StartExecution chỉ trả về executionArn và startDate và không thể thay đổi đầu ra (default behavior).
  + Hướng giải quyết 1: Tạo thêm một method GET với action DescribeExecution để phía backend gọi liên tục đều đặn để xem statemachine hoành thành quá trình hay chưa => Không đúng theo yêu cầu quá trình chạy đồng bộ (asynchronous)
  + Hướng giải quyết 2: Tạo thêm một Statemachine loại Express, hoạt động như wrapper để kích hoạt Statemachine trong khi trả về kết quả thông qua StartSyncExecution trong cùng 1 API, trong khi Statemachine phía trong vẫn chạy.

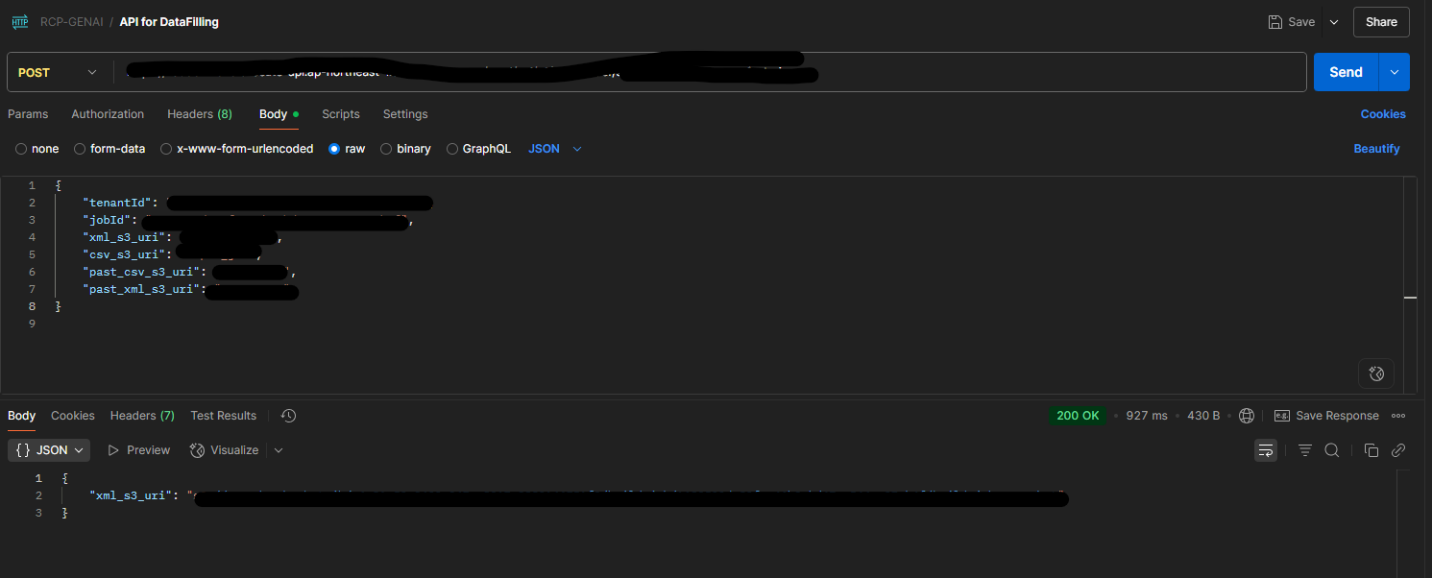


Hình . Trao đổi với anh David Le về xử lý Step Function

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Step function sau hướng giải quyết 1



Hình . Step function sau hướng giải quyết 2

#### Commit trên GitLab

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Commit trên nhánh feature/basic-information-conversion

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Commit trên nhánh feature/similar-job-searching

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Commit trên nhánh feature/data-filling

# Chương 5: Tổng kết

## Tổng kết:

* Hoàn thành tất cả những nhiệm vụ được giao, đúng tiến độ
* Mang lại những đóng góp tích cực cho nhóm và dự án.

## Thuận lợi:

* Môi trường làm việc năng động, tích cực, thân thiện và rất vui vẻ nhưng vẫn nghiêm túc cần có trong công việc.
* Nhiều kiến thức nền tảng được học ở trường có hiệu quả khi áp dụng ở trong môi trường doanh nghiệp.
* Được các anh chị và đồng nghiệp hỗ trợ và giúp đỡ tận tình.

## Khó khăn gặp phải:

* Ở giai đoạn đầu khi em tham gia dự án, điều khó khăn nhất mà bản thân em gặp phải là khi bản thân phải tự nghiên cứu mọi thứ về hệ thống và mã nguồn. Hệ thống lớn và phức tạp, nên việc đọc hiểu thiết kế hệ thống, cấu trúc lưu trữ, thành phần API tốn nhiều thời.
* Làm việc với dự án của người nước ngoài gặp khó khăn trong rào cản ngôn ngữ, source code của khách được chú thích nhiều bằng tiếng Nhật cũng như cái màn hình front end, dummy data. Dẫn đến khó khăn trong quá trình kiểm thử kết quả đầu ra đúng hay không.
* Do thiếu kinh nghiệm trong việc tiếp cận nền tảng cloud Amazon Web Services, việc kết hợp các services AWS lại với nhau gặp khó khăn, một phần do mỗi người trong nhóm chịu trách nhiệm 3 dịch vụ khác nhau. Không hiểu cấu trúc hoạt động của từng dịch vụ để kết hợp lại với nhau.

## Đóng góp cho công ty:

* Em đã hoàn thành hết các task được giao, dù có 1-2 task bị trễ thời gian dự tính nhưng không gây ảnh hưởng lớn tới tiến độ của nhóm.
* Tham gia đóng góp ý kiến trong quá trình tham gia dự án thực tế cho nhóm.
* Hỗ trợ các anh chị trong bằng cách kiểm thử chất lượng, thay đổi theo yêu cầu của khách.
* Áp dụng được các kiến thức đã học ở trường vào để hoàn thành các ticket trong thời gian ngắn nhất.

# Tài liệu tham khảo

|  |  |
| --- | --- |
| [1] | "AWS Lambda developer guide," Amazon Web Services, [Online]. Available: https://docs.aws.amazon.com/lambda/latest/dg/welcome.html. |
| [2] | "AWS API Gateway developer guide," Amazon Web Services, [Online]. Available: https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html. |
| [3] | "AWS Lambda API reference," Amazon Web Services, [Online]. Available: https://docs.aws.amazon.com/lambda/latest/api/API\_Operations.html. |
| [4] | "AWS Step Functions developer guide," Amazon Web Services, [Online]. Available: https://docs.aws.amazon.com/step-functions/latest/dg/statemachine-structure.html. |
| [5] | "AWS Step Functions API reference," Amazon Web Services, [Online]. Available: https://docs.aws.amazon.com/step-functions/latest/apireference/API\_Operations.html. |