NEXLAB INTERNSHIP PROGRAM 2025 HO CHI MINH

NEXLAB TECHNOLOGY

Assessment:

Workshop 1

Mentor(s): Chiến Phạm Minh
Anh Nguyen M.Sc
Bang Vo M.Sc

Intern(s): Trương Tiến Anh 22120017

Ho Chi Minh City, 7/2025

LỜI MỞ ĐẦU

Báo cáo này trình bày tóm tắt buổi workshop, mục tiêu để kiểm tra kiến thức đã học và khả năng giải quyết vấn đề thực tế, từ đó giúp công ty đánh giá sự phù hợp và tiềm năng phát triển của ứng viên.
Bài giảng đã giúp em hiểu rõ hơn những kiến thức cần thiết cho chương trình thực tập, đồng thời nâng cao kỹ năng tư duy logic và cách trình bày giải pháp một cách rõ ràng, mạch lạc.
Em xin chân thành cảm ơn công ty NEXLAB đã tạo cơ hội học hỏi và đánh giá năng lực, đồng thời cảm ơn các anh/chị đã hướng dẫn, hỗ trợ em trong suốt quá trình này.

Trương Tiến Anh
Sinh viên Khoa Khoa học Máy tính & Khoa học Dữ liệu
Trường Đại học Khoa học Tự nhiên, ĐHQG TP. Hồ Chí Minh
TP. Hồ Chí Minh, 7/2025

Mục lục

Phần I
Giới thiệu và tiến độ công việc

Thông tin ứng viên:

STT	Họ Tên	MSSV	Email
1	Trương Tiến Anh	22120017	truongtienanh16@gmail.com

Bảng 1: *Thông tin ứng viên*

Công việc được giao và tiến độ công việc:

STT	Công việc được giao	Tiến độ công việc	Vấn đề gặp phải
1	Tóm tắt buổi workshop 1	Hoàn thành	Không có

Bảng 2: *Công việc*

Phần II
Báo cáo chi tiết

1 Mục tiêu & Tổng quan Workshop

1.1 Mục tiêu

- Trang bị kiến thức thực tiễn về Data, AI, Backend, DevOps cho sinh viên.
- Nâng cao tư duy triển khai sản phẩm AI, Data thực tế trong doanh nghiệp.
- Rèn luyện kỹ năng teamwork, báo cáo, tự học và giải quyết vấn đề thực tế.

1.2 Định hướng chương trình

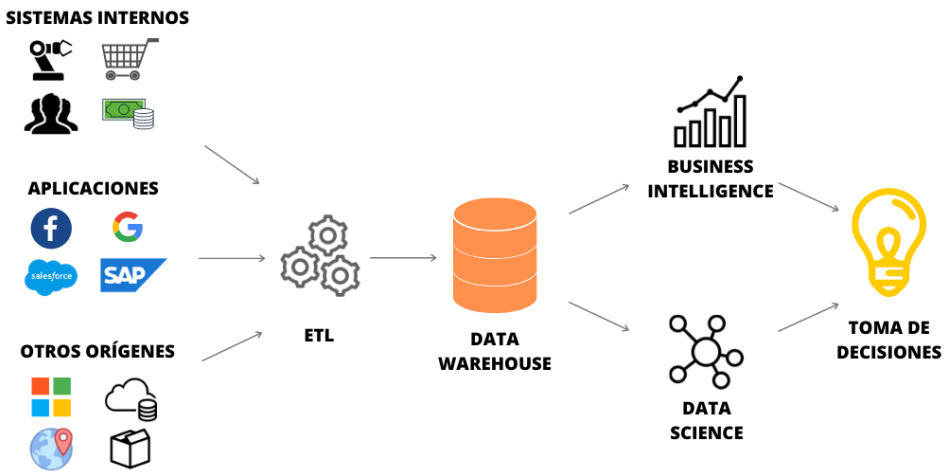
- Kết hợp kiến thức về Data Engineering, AI/ML, Backend, Frontend, DevOps.
- Làm dự án thực tế: AI Agent, Computer Vision, tích hợp backend, frontend, data pipeline.

2 Data & Data Warehouse

2.1 Data – Nền tảng của AI & IT

- **Khái niệm**: Dữ liệu (data) là nguyên liệu đầu vào bắt buộc cho mọi hệ thống AI, ML, BI.
- **Vai trò**: Model AI dù tốt đến đâu cũng cần data chất lượng, đủ lớn, đa dạng, liên tục cập nhật.
- **Tính chất data tốt**: Đầy đủ, sạch, nhất quán, đúng định dạng, có nhãn (label nếu làm supervised learning).

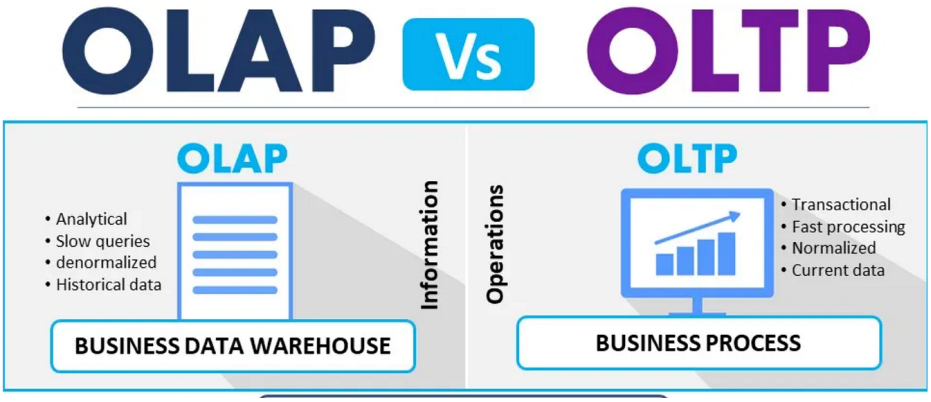
2.2 Data Warehouse (DWH)



Hình 1: *Data warehouse*

- **Định nghĩa**: Hệ thống kho lưu trữ tập trung, tích hợp dữ liệu từ nhiều nguồn trong doanh nghiệp (app, web, DB, file, IoT...).
- Chức năng:
  - Tập trung hóa dữ liệu, phục vụ truy vấn tổng hợp, báo cáo, dashboard, phân tích nghiệp vụ.
  - Lưu dữ liệu lịch sử, phục vụ phân tích xu hướng, dự báo.

2.3 OLTP vs OLAP



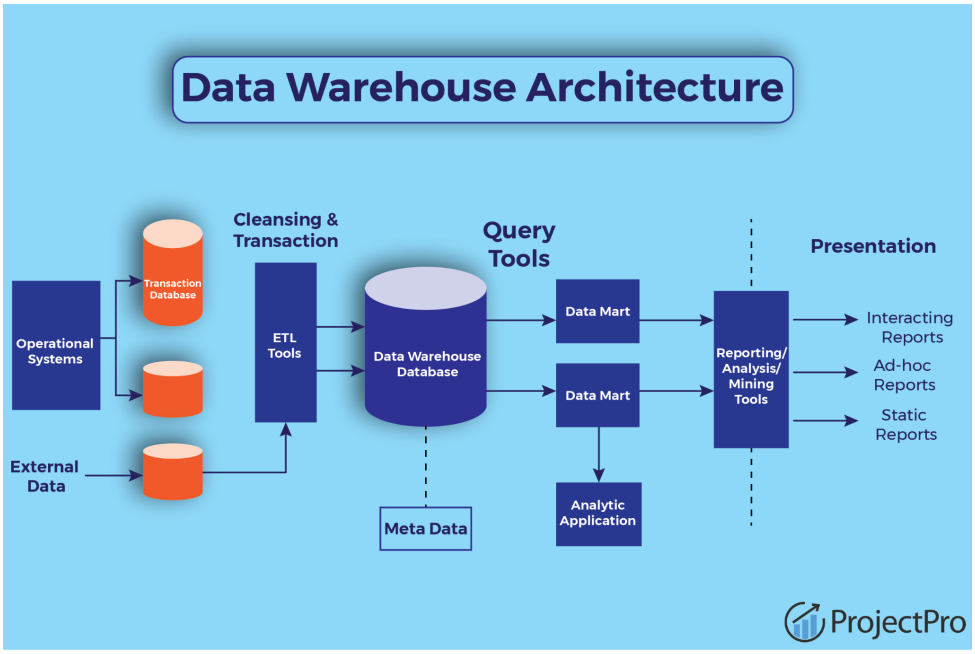
Hình 2: *OLTP vs OLAP*

- OLTP (Online Transaction Processing):
  - Mục đích: Xử lý giao dịch thường ngày (mua hàng, đăng ký tài khoản, cập nhật thông tin...).
  - Dữ liệu: Chuẩn hóa mạnh (1NF, 2NF, 3NF), giảm trùng lặp, đảm bảo toàn vẹn, nhất quán.
  - Truy vấn: Nhỏ, đơn lẻ, nhanh (insert/update/delete nhanh).
  - Ví dụ: Hệ thống database cho app bán hàng, web app.
- OLAP (Online Analytical Processing) / DWH:
  - Mục đích: Phân tích tổng hợp, báo cáo, phân tích dữ liệu lớn (big data).

  - Dữ liệu: Ngược chuẩn hóa (denormalized) để tăng tốc truy vấn tổng hợp (group by, sum, count...).
  - Truy vấn: Lớn, phức tạp, tổng hợp, có thể chấp nhận chậm hơn một chút.
  - Ví dụ: Truy vấn doanh thu theo ngày, theo tháng, top sản phẩm bán chạy.

- Tại sao không dùng OLTP cho phân tích?
  - Truy vấn tổng hợp trên DB OLTP sẽ chậm, ảnh hưởng hiệu suất hệ thống vận hành.
  - Dữ liệu phân mảnh, phải join nhiều bảng dẫn tới chậm, khó mở rộng.

2.4 Kiến trúc tổng thể Data Warehouse



Hình 3: *DWH Architecture*

- Các tầng chính:
  - 1. Data Sources: DB, file, API, IoT, app, CRM, ERP...
  - 2. Staging/Integration Layer: Nơi gom dữ liệu tạm, xử lý ETL/ELT.
  - 3. Data Warehouse: Kho tập trung, dữ liệu đã chuẩn hóa lại, tối ưu cho truy vấn tổng hợp.
  - 4. Data Mart: Phân vùng dữ liệu cho từng phòng ban (Sale, HR, Marketing...).

- ETL/ELT
  - Extract: Kết nối, lấy dữ liệu từ các nguồn khác nhau.
  - Transform: Làm sạch, chuẩn hóa, convert format, tạo chỉ số, tổng hợp, chuyển về dạng dễ phân tích.
  - Load: Đưa dữ liệu vào DWH, tối ưu phân mảnh (theo ngày, tháng...), lưu trữ hiệu quả cho truy vấn.

- Kiến trúc Medallion (Bronze/Silver/Gold):
  - Bronze: Dữ liệu thô, chưa xử lý, giữ nguyên nguồn.
  - Silver: Đã làm sạch, chuẩn hóa, convert format (ví dụ: chuẩn hóa ngày, lọc dữ liệu lỗi).
  - Gold: Đã tổng hợp, tạo ra bảng phân tích, tổng hợp chỉ số sẵn sàng cho dashboard.

2.5 Đặc trưng & tiêu chí của Data Warehouse

- Tập trung hóa: Dữ liệu từ nhiều nguồn quy về một nơi, dễ quản lý, hạn chế trùng lặp.
- Đầy đủ lịch sử: Có thể lưu toàn bộ dữ liệu từ đầu, phục vụ phân tích dài hạn.
- Tích hợp đa nguồn: Kết nối đa dạng (DB, file, API, streaming, IoT...).
- Phục vụ phân tích, không phục vụ giao dịch: Tối ưu hóa cho query tổng hợp, dashboard, báo cáo, BI.
- Bảo mật & phân quyền: Chỉ những nhóm người dùng/phòng ban nhất định xem được dữ liệu phù hợp.
- Mở rộng linh hoạt: Dễ scale, thêm nguồn, thêm mục đích sử dụng, phân chia Data Mart.

2.6 Xu hướng hiện đại (so sánh truyền thống - hiện đại)

- DWH truyền thống: Triển khai on-premise, chi phí lớn, khó mở rộng, cập nhật chậm.
- DWH cloud: AWS Redshift, GCP BigQuery, Snowflake, Databricks... - dễ mở rộng, chi phí linh hoạt, tích hợp AI/ML, real-time.
- Data Lake, Lakehouse: Hỗ trợ lưu trữ dữ liệu phi cấu trúc (ảnh, video, JSON, log...), giảm chi phí, tăng khả năng phân tích đa dạng.
- Orchestration, Data Catalog, Data Governance: Tự động hóa pipeline, quản lý dữ liệu, phân quyền, theo dõi lineage.

3 Kỹ năng Backend, DevOps, Web cho AI Engineer

3.1 DevOps & Backend

- Git:
  - Quản lý version code, branch, merge, conflict.
  - Workflow: development → staging → production, review code, pull request.
- Docker vs VMware:
  - Docker: Container hóa, đóng gói app, dùng chung nhân OS, tối ưu tài nguyên, CI/CD nhanh.
  - VMware: Máy ảo riêng biệt, có hệ điều hành riêng, bảo mật tốt hơn nhưng nặng, scale kém.
- Docker Compose: Quản lý nhiều container (web, DB, cache...), dễ dàng build/deploy môi trường thực tế, testing.

3.2 Web Backend (Python)

- Framework
  - FastAPI: Viết REST API, tích hợp dễ với AI model, async support tốt, dễ học.
  - Flask: Đơn giản, dễ mở rộng, nhiều tutorial.
- Cơ chế web server:
  - WSGI/ASGI: Chuẩn giao tiếp giữa web server và Python app (ASGI hỗ trợ async).
  - Process/thread: Hiểu cách chia nhỏ xử lý trên server, tối ưu concurrency.
  - Middleware: Xử lý chung cho nhiều API (logging, auth, error handling...).
- Tổ chức code:
  - Tách rõ controller (route/API), service (logic xử lý), model/schema (DB), config, test.
  - Dễ maintain, scale, chuyển giao team.

3.3 Frontend

- Framework:
  - ReactJS: Component hóa, state management, dễ mở rộng.
  - NextJS: SSR, tối ưu SEO, render nhanh, chia nhỏ route/module.
- Kết nối backend: Gọi REST API, xử lý response, quản lý state (Redux, Context...).

3.4 Database

- RDBMS (quan hệ):
  - PostgreSQL: Truy vấn mạnh mẽ, đảm bảo ACID, phù hợp dữ liệu logic, structured.
  - Ưu điểm: Hỗ trợ join, aggregate, constraint, index.
- NoSQL
  - MongoDB: Lưu trữ linh hoạt, phù hợp dữ liệu phi cấu trúc, schema động.
  - Ưu điểm: Dễ scale, lưu JSON, insert/update nhanh.
- Vector database:
  - Lưu embedding (AI), phục vụ search QA, chatbot, agent.
  - Dùng local cho demo, cloud cho production.
- ORM: SQLAlchemy, Tortoise... giúp map object <-> DB, query thuận tiện, giảm lỗi, dễ maintain.

Phần III
Tài liệu tham khảo

[1] Data warehouse
URL: https://topdev.vn/blog/data-warehouse-la-gi-tong-quan-ve-kho-du-lieu/

[2] OLTP vs OLAP
URL: https://viblo.asia/p/oltp-va-olap-co-gi-khac-nhau-maGK786BZj2.

[3] Docker
URL: https://topdev.vn/blog/docker-la-gi/.