Handover Document – Project Pathwise

1. Executive Summary

Pathwise là một hệ thống quản lý toàn diện dành cho nhân viên, hỗ trợ theo dõi và phát triển kỹ năng, mục tiêu học tập, nhận và quản lý feedback, cũng như đề xuất các khóa học phù hợp. Hệ thống được thiết kế để có thể mở rộng, dễ bảo trì và tích hợp với các dịch vụ bên ngoài.

Key Features:

* Quản lý kỹ năng (Skills) và mapping giữa các kỹ năng và mục tiêu nghề nghiệp.
* Quản lý mục tiêu học tập (Goals) của nhân viên.
* Thu thập, lưu trữ và phân tích phản hồi (Feedback).
* Đề xuất khóa học tự động dựa trên skill embedding và external course API.
* Hỗ trợ tích hợp với các nền tảng bên ngoài như Coursera.

Stack kỹ thuật:

* Backend: Python 3.11+, FastAPI, SQLAlchemy ORM.
* Frontend: ReactJS, quản lý state bằng Redux hoặc Context API.
* Database: PostgreSQL (AWS RDS), Redis cache.
* Vector Store: Milvus để lưu embedding kỹ năng.
* Infrastructure: Docker, Docker Compose, Kubernetes, triển khai trên AWS (ECR, Secrets Manager, RDS, S3).

2. Repository Structure

Pathwise/

│

├─ main.py # Entry point backend, chạy ứng dụng FastAPI

├─ requirements.txt # Dependencies Python

├─ Dockerfile # Docker image build

├─ docker-compose.yml # Docker services (backend, Milvus, Redis)

├─ deployment.yaml # Kubernetes deployment config

├─ service.yaml # Kubernetes service config

│

├─ core/ # Cấu hình chung (logging, security, utils)

├─ database/ # ORM models, migration scripts

├─ schemas/ # Pydantic schemas cho API input/output

│

├─ api/ # Route endpoints theo module:

│ ├─ auth.py # Xác thực người dùng, JWT

│ ├─ feedback.py # CRUD feedback

│ ├─ goal.py # CRUD mục tiêu học tập

│ ├─ recommendation.py # Endpoint gợi ý khóa học

│ └─ external\_courses.py # Lấy dữ liệu khóa học từ nền tảng bên ngoài

│

├─ apps/ # Business logic chia module rõ ràng

│ ├─ skill\_management/ # Xử lý mapping, scoring kỹ năng

│ ├─ goal\_tracking/ # Logic tính tiến độ, gợi ý hành động

│ └─ recommendation\_engine/ # Embedding, tìm khóa học phù hợp

│

├─ front\_end/ # ReactJS app

│ ├─ src/

│ ├─ public/

│ └─ package.json

│

└─ skills\_cache.json # Cache embedding kỹ năng, giảm tần suất query Milvus

3. Environment & Secrets

Môi trường phát triển & triển khai:

* Python >= 3.11, NodeJS >= 18, Docker >= 23
* Cấu hình qua .env:
* # Coursera API
* URI\_COURSERA=https://api.coursera.org/api/courses.v1
* # API embedding
* EM\_KEY\_EMBEDDING\_1=your-key-1
* EM\_KEY\_EMBEDDING\_2=your-key-2
* EM\_KEY\_EMBEDDING\_3=your-key-3
* EM\_KEY\_EMBEDDING\_4=your-key-4
* # RDS
* RDS\_ENDPOINT=your-rds-endpoint
* RDS\_PORT=5432
* RDS\_USER=your-db-user
* RDS\_PASSWORD=your-db-password
* RDS\_DATABASE=your-db-name
* # Milvus
* MIL\_HOST=your-milvus-host
* MIL\_PORT=19530
* # GenAI
* GEN\_KEY\_GEN\_1=your-genai-key
* # O2Auth
* SECRET\_KEY=your-secret-key
* ALGORITHM=HS256

Secrets management:

* AWS Secrets Manager: lưu IAM, DB credentials, API keys.
* Không commit secrets vào git, sử dụng .env.example để chia sẻ mẫu cấu hình.
* External services API keys (Coursera, internal APIs) được load qua env variables.

Best practice:

* Kiểm tra .env trước khi deploy.
* Sử dụng IAM Role với EC2/K8s để tránh lưu secret trực tiếp.

4. Data Model

ORM & Schema:

* SQLAlchemy cho các bảng chính:
  + User: Thông tin nhân viên, role, status.
  + Skill: Kỹ năng, level, embedding vector.
  + Goal: Mục tiêu học tập, deadline, trạng thái.
  + Feedback: Nhận xét, đánh giá, timestamp.
  + Recommendation: Khóa học gợi ý, lý do, liên kết.
  + ExternalCourse: Khóa học từ Coursera/LinkedIn Learning, metadata.

Schema management:

* Pydantic schemas (schemas/) đảm bảo dữ liệu đầu vào/out đúng chuẩn, có validation.
* Module-specific schemas giúp backend dễ mở rộng.

Sample data & migrations:

* data/: SQL scripts để seed DB.
* Migration scripts: hỗ trợ cập nhật DB mà không mất dữ liệu.

5. Vector Store

* Milvus lưu trữ embedding kỹ năng để phục vụ việc gợi ý khóa học thông minh.
* Embedding model: text-embedding-004 (Gemini).
* skills\_cache.json: giảm truy vấn Milvus cho các skills thường dùng, tăng performance.

Lưu ý:

* Khi thêm skill mới, cần update embedding và cache.
* Sử dụng batch insert để tối ưu Milvus performance.

6. API Surface

FastAPI với OpenAPI docs: /docs

Các route chính:

| Endpoint | Function |
| --- | --- |
| /api/auth | Đăng nhập, đăng ký, refresh JWT |
| /api/feedback | Gửi & lấy feedback, phân loại theo user/goal |
| /api/goal | Quản lý mục tiêu học tập, update tiến độ |
| /api/recommendation | Lấy khóa học gợi ý dựa trên kỹ năng |
| /api/external\_courses | Lấy dữ liệu khóa học từ bên ngoài |
| /api/skills\_mapping | Mapping kỹ năng & scoring |

Best practice:

* Các endpoint tuân thủ RESTful conventions.
* Validation input bằng Pydantic.
* Logging & exception handling đã được tích hợp.

7. Frontend

* ReactJS app (front\_end/)
* API endpoint config: src/apiConfig.js
* Component chính: Dashboard, Feedback, Goal, Recommendation
* Build commands:

npm install

npm run build

* Khuyến nghị: sử dụng ESLint & Prettier để duy trì code style.

8. Runbook (Chạy local)

1. Cài đặt môi trường: Python, NodeJS, Docker.
2. Tạo .env theo mẫu .env.example.
3. Chạy Milvus:

cd /mnt/data\_volume/milvus

docker compose up -d

1. Chạy backend:

uvicorn main:app --reload

1. Chạy frontend:

cd front\_end

npm start

1. Truy cập: http://localhost:8000 (backend) và http://localhost:3000 (frontend).

9. Deployment trên AWS

* Docker image: Push lên ECR pathwise-hcm-1.0.0
* Database: AWS RDS PostgreSQL
* Secrets: AWS Secrets Manager
* Kubernetes:
  + deployment.yaml, service.yaml
  + Cập nhật env & secrets khi deploy
* Best practice:
  + CI/CD pipelines (GitHub Actions, CodePipeline)
  + Logging & monitoring (CloudWatch, Prometheus/Grafana)

10. Final Model for Next Dev

* Code: Module rõ ràng, dễ mở rộng.
* Schemas & API: Đầy đủ, có ví dụ dữ liệu.
* Security: Secrets được bảo mật, endpoint kiểm tra kỹ input.
* Recommendations for future:
  + CI/CD để tự động build, test, deploy.
  + Bổ sung unit & integration tests.
  + Logging, monitoring & alerting cho production.
  + Hợp nhất caching strategy để tăng performance cho API & embedding lookup.