

Phần 1: Tầm nhìn và Mục tiêu Chiến lược

1.1 Tuyên bố Sứ mệnh (Mission Statement)

Mục tiêu của dự án CVFlow là thiết kế, xây dựng và triển khai một "Hệ thống Theo dõi Ứng viên" (Applicant Tracking System - ATS) thông minh, với triết lý cốt lõi là "Python-centric" (lấy Python làm trọng tâm).

Sứ mệnh của dự án này không phải là xây dựng một ứng dụng web toàn diện với giao diện phức tạp. Thay vào đó, sứ mệnh là chứng minh năng lực làm chủ các kỹ thuật xử lý tệp (File Processing), Xử lý Ngôn ngữ Tự nhiên (Natural Language Processing - NLP) và học máy cơ bản bằng Python. Chúng ta sẽ xây dựng một pipeline xử lý dữ liệu backend mạnh mẽ để tự động hóa và thông minh hóa các tác vụ cốt lõi trong quy trình quản lý tuyển dụng.

1.2 Vấn đề Nghiệp vụ (The Business Problem)

Trong bối cảnh tuyển dụng hiện đại, bộ phận Nhân sự (HR) phải đối mặt với một khối lượng lớn hồ sơ ứng tuyển (CV) cho mỗi vị trí. Quy trình sàng lọc thủ công, bao gồm việc đọc từng CV, trích xuất thông tin liên hệ, tìm kiếm kỹ năng, và so sánh với Mô tả Công việc (Job Description - JD), là cực kỳ tốn thời gian, không nhất quán, và dễ dẫn đến bỏ sót ứng viên tiềm năng do lỗi chủ quan của con người.

1.3 Giải pháp Kỹ thuật (The Technical Solution)

CVFlow sẽ giải quyết vấn đề này bằng cách cung cấp một pipeline xử lý dữ liệu tự động, được kích hoạt ngay khi ứng viên nộp CV (dưới dạng tệp .pdf hoặc .docx). Hệ thống backend, được xây dựng hoàn toàn bằng Python, sẽ thực hiện các bước sau:

- Tự động Bóc tách (File Processing):** Đọc và trích xuất 100% nội dung văn bản thô (raw text) từ các tệp CV.
- Trích xuất Thông tin (NER):** Áp dụng các kỹ thuật NLP (regex, NLTK) để tự động nhận dạng và trích xuất các thực thể quan trọng (Tên, Email, SĐT, danh sách Kỹ năng).
- Tóm tắt Thông minh (Summarization):** Tự động tạo bản tóm tắt súc tích cho cả CV (giúp HR đọc nhanh) và JD (giúp ứng viên nắm bắt nhanh yêu cầu).
- Chấm điểm & So khớp (Scoring & Matching):** Tự động chấm điểm mức độ phù hợp của CV với JD dựa trên các kỹ năng có trọng số và áp dụng các thuật toán so khớp ngữ nghĩa (TF-IDF, Cosine Similarity).

Giao diện người dùng (UI) và Cơ sở dữ liệu (CSDL) sẽ được giữ ở mức tối giản bắt buộc, chỉ đóng vai trò là phương tiện để "đưa dữ liệu vào" (upload) và "trình bày kết quả" (dashboard) từ

pipeline Python.

1.4 Các Yếu tố Thành công Then chốt (Critical Success Factors - CSFs)

Sự thành công của đề án này sẽ được đo lường dựa trên các yếu tố sau:

- **CSF 1 (Trọng tâm):** Pipeline 5 module Python của MVP (Phần 2.2) hoạt động ổn định, chính xác, và được tích hợp đầy đủ (end-to-end) từ API đến CSDL.
- **CSF 2 (Tiến độ):** Hoàn thành và triển khai thành công MVP (với 5 module) trong vòng 10 tuần, theo đúng Lộ trình (Phần 6).
- **CSF 3 (Chất lượng):** 100% mã nguồn Python trong nhánh `main` tuân thủ tuyệt đối các tiêu chuẩn chất lượng (định dạng bằng `black`, kiểm lỗi bằng `flake8`) được định nghĩa trong Phần 5.
- **CSF 4 (Hợp tác):** Mỗi thành viên trong nhóm đều có đóng góp mã nguồn Python ý nghĩa vào các module nghiệp vụ, tuân thủ triết lý "Sở hữu Chung" (Phần 4).

Phần 2: Phạm vi Dự án và Kiến trúc Module Python

2.1 Triết lý Phạm vi: "Pipeline-First"

Phạm vi của dự án CVFlow được cố định và tập trung tuyệt đối vào pipeline xử lý dữ liệu backend. Mọi nỗ lực phát triển, phân bổ thời gian và nguồn lực đều phải ưu tiên cho việc xây dựng và hoàn thiện các module Python. Giao diện (UI) và Cơ sở dữ liệu (CSDL) sẽ được giữ ở mức tối giản bắt buộc như định nghĩa trong "Phạm vi Loại trừ" (2.4). Mọi hành vi "mở rộng" hoặc "phức tạp hóa" giao diện và CSDL đều được xem là "scope creep" (vượt ngoài phạm vi) và bị cấm.

2.2 Sản phẩm Khả dụng Tối thiểu (MVP)

MVP là một hệ thống hoạt động đầy đủ, nơi một Quản trị viên (HR) có thể tạo "Vị trí Tuyển dụng" (Job), một ứng viên (public) có thể nộp CV, và HR có thể xem danh sách ứng viên đã được hệ thống tự động xử lý, trích xuất thông tin, tóm tắt, chấm điểm và so khớp.

2.2.1 Quản lý Căn bản (Backend)

- **CSDL (SQLite):** Thiết kế CSDL tối giản với các bảng chính: `Job` (id, title, description), `Candidate` (id, name, email, phone), `CV_File` (id, candidate_id, file_path, raw_text), `Extracted_Skill` (id, candidate_id, skill_name), `Score` (candidate_id, job_id, score_value).
- **API (Flask):** Các API endpoint cơ bản:
 - `POST /job` (cho HR tạo Job mới, lưu JD vào CSDL).

- GET /jobs (cho Public UI lấy danh sách Job).
- GET /job/<id> (cho Public UI lấy chi tiết Job, bao gồm bản tóm tắt NLTK của JD).
- POST /apply (cho Public UI upload CV, kích hoạt toàn bộ pipeline xử lý Python).
- GET /admin/candidates (cho Admin UI xem danh sách ứng viên đã được xử lý).

2.2.2 Giao diện (Frontend Tối giản)

- **Public UI:** Giao diện web tối giản (khuyến khích dùng template HTML/CSS):
 - Trang 1: Hiển thị danh sách các Job đang tuyển.
 - Trang 2: Chi tiết Job (hiển thị bản tóm tắt NLTK của Job) và một form nộp CV (chấp nhận .pdf / .docx).
- **Admin UI:** Một dashboard (bảng điều khiển) tối giản:
 - Giao diện để POST Job mới.
 - Giao diện xem danh sách ứng viên theo từng Job.
 - Giao diện xem chi tiết ứng viên, hiển thị rõ ràng: thông tin đã trích xuất (Module 2), bản tóm tắt CV (Module 3), điểm chấm (Module 4), và điểm so khớp (Module 5).

2.2.3 Các Module Python Cốt lõi (Trái tim của CVFlow)

Đây là trọng tâm tuyệt đối của dự án.

- **Module Python 1: Xử lý và Trích xuất Văn bản (File Processing)**
 - **Mô tả:** Module này là bước đầu tiên của pipeline, nhận đường dẫn tệp (path) của CV làm đầu vào và chịu trách nhiệm bóc tách 100% văn bản thô (raw text).
 - **Công nghệ:** Sử dụng thư viện python-docx (cho .docx) và PyPDF2 (cho .pdf).
 - **Kết quả:** Trả về là một chuỗi (string) văn bản thô duy nhất.
- **Module Python 2: Trích xuất Thực thể (NER - Named Entity Recognition)**
 - **Mô tả:** Module này nhận văn bản thô từ Module 1, áp dụng các kỹ thuật NLP cơ bản để tự động trích xuất các thông tin có cấu trúc và bán cấu trúc.
 - **Công nghệ:** Sử dụng regex (Biểu thức Chính quy) của Python cho các mẫu cố định (Email, Số điện thoại). Sử dụng NLTK (ví dụ: Part-of-Speech tagging, chunking) hoặc một danh sách từ khóa kỹ năng (keyword dictionary) để xác định và trích xuất Tên (Name) và Kỹ năng (Skills).
- **Module Python 3: Module Tóm tắt Tự động (NLTK - Tóm tắt Trích xuất)**
 - **Mô tả:** Xây dựng một module tóm tắt trích xuất (Extractive Summarization) dựa trên thuật toán thống kê. Module này tính toán các câu quan trọng nhất dựa trên tần suất từ (sau khi loại bỏ stop-words) hoặc một biến thể đơn giản của thuật toán TextRank/LexRank.
 - **Công nghệ:** Sử dụng NLTK (cho tokenization, stop-words removal, sentence splitting).

- **Yêu cầu Kiến trúc (CTO Insight):** Module này phải được thiết kế với một interface (giao diện lập trình) trừu tượng rõ ràng, ví dụ: `class BaseSummarizer` với phương thức `summarize(text)`. Phiên bản MVP sẽ triển khai `class NLTKSummarizer(BaseSummarizer)`. Thiết kế này cho phép chúng ta "hoán đổi" (swap) với module AI nâng cao (Phần 2.3) trong tương lai mà không cần thay đổi code ở tầng API.
- **Module Python 4: Chấm điểm CV (Scoring)**
 - **Mô tả:** Một module logic Python thuần túy. Khi HR tạo Job (API `POST /job`), họ có thể định nghĩa các "trọng số" cho kỹ năng (ví dụ: "Python": 10, "Java": 5, "SQL": 8). Module này sẽ so sánh danh sách kỹ năng trích xuất từ CV (Module 2) với các kỹ năng yêu cầu của Job và tính toán % phù hợp.
 - **Công nghệ:** Logic `if/else` và các phép toán học cơ bản của Python.
- **Module Python 5: So khớp Thông minh (Matching)**
 - **Mô tả:** Module này cung cấp khả năng so khớp ngữ nghĩa (semantic matching) giữa toàn bộ văn bản CV và toàn bộ văn bản Mô tả Job, thay vì chỉ dựa vào từ khóa (như Module 4).
 - **Công nghệ:** Sử dụng thư viện `scikit-learn`. Pipeline chuẩn sẽ được áp dụng:
 1. Tạo một kho văn bản (corpus) gồm (văn bản CV + văn bản Job).
 2. Vector hóa văn bản bằng `TfidfVectorizer` (Term Frequency-Inverse Document Frequency) để chuyển văn bản thành các vector số học.
 3. Tính toán `cosine_similarity` (độ tương đồng cosine) giữa vector CV và vector Job để đưa ra điểm số so khớp (từ 0 đến 1).

2.3 Mục tiêu Mở rộng (Stretch Goals - Tích hợp AI Nâng cao)

Đây là các mục tiêu chỉ được phép thực hiện sau khi MVP (2.2) đã hoàn thành 100% (theo DoD - Phần 5) và đã được triển khai.

- **Nâng cấp Module 3 (AI - Tóm tắt Trừu tượng)**
 - **Mô tả:** Nghiên cứu và tích hợp một mô hình AI đã được huấn luyện trước (pre-trained) để thực hiện tóm tắt trừu tượng (Abstractive Summarization), tức là mô hình tự "viết" lại các câu tóm tắt mới thay vì chỉ trích xuất câu gốc.
 - **Công nghệ:** Tích hợp thư viện `Hugging Face Transformers` (với `torch`). Tạo một class mới: `class TransformerSummarizer(BaseSummarizer)` (ví dụ: sử dụng mô hình T5-small hoặc PEGASUS) và triển khai interface đã định nghĩa trong Module 3 (MVP).
- **Module Python 6: Phát hiện Trùng lặp (Deduplication)**
 - **Mô tả:** Khi HR xem danh sách ứng viên, hệ thống sẽ chạy một quy trình nền để cảnh báo nếu có các ứng viên có khả năng bị trùng lặp (ví dụ: "Nguyen Van A" và "A. Nguyen Van" nộp 2 lần).

- **Công nghệ:** Sử dụng thư viện `fuzzywuzzy` (Python) để thực hiện so khớp mờ (fuzzy matching) tên ứng viên (ví dụ: `fuzz.token_set_ratio`).

2.4 Phạm vi Loại trừ (Out of Scope)

Để đảm bảo 100% nguồn lực tập trung vào triết lý "Python-centric", dự án này sẽ **KHÔNG** thực hiện và **NGHIÊM CẤM** các hạng mục sau:

- **Giao diện Phức tạp:** Bất kỳ nỗ lực nào sử dụng các framework JavaScript frontend nặng (ví dụ: React, Vue, Angular), các dashboard kéo-thả, hoặc thiết kế responsive (thích ứng) phức tạp.
- **Xác thực Phức tạp:** Hệ thống xác thực và phân quyền người dùng phức tạp (ví dụ: OAuth, JWT, phân quyền đa cấp). Một mật khẩu admin đơn giản (hard-coded hoặc biến môi trường) là đủ cho Admin UI.
- **CSDL Nặng nề:** Sử dụng các hệ quản trị CSDL quan hệ cần cài đặt và quản trị (ví dụ: PostgreSQL, MySQL). `SQLite` là bắt buộc cho MVP.
- **Quy trình Phức tạp:** Quản lý phiên (session) nâng cao hoặc quy trình tuyến dụng nhiều bước (multi-step workflow).

Phần 3: Ngăn xếp Công nghệ và Công cụ (Tech Stack)

Phần này định nghĩa các công nghệ và công cụ được phê duyệt cho dự án. Mọi công nghệ ngoài danh sách này phải được CTO/PM duyệt trước khi sử dụng.

3.1 Backend

- **Ngôn ngữ:** `Python 3.9+`
- **Framework:** `Flask` (Lý do: Nhẹ, linh hoạt, lý tưởng cho việc xây dựng API và tích hợp các module Python, không ép buộc cấu trúc nặng nề).

3.2 NLP / Xử lý Dữ liệu (MVP)

- **Xử lý tệp:** `python-docx`, `PyPDF2`
- **Xử lý NLP cơ bản:** `NLTK` (cho tokenization, stop-words, POS tagging), `regex`
- **Machine Learning (Matching):** `scikit-learn` (cho `TfidfVectorizer`, `cosine_similarity`), `numpy` (hỗ trợ tính toán).

3.3 NLP / Xử lý Dữ liệu (Mục tiêu Mở rộng)

- **AI (Tóm tắt):** `transformers` (Hugging Face), `torch`
- **So khớp mờ:** `fuzzywuzzy`

3.4 Cơ sở dữ liệu

- **MVP:** SQLite (Lý do: Tích hợp sẵn trong Python, không cần cài đặt, serverless, loại bỏ hoàn toàn gánh nặng quản trị CSDL, cho phép nhóm tập trung 100% vào logic Python).

3.5 Frontend (Tối giản)

- **Ngôn ngữ:** HTML5, CSS3, JavaScript (ES6+)
- **Yêu cầu:** Khuyến khích mạnh mẽ việc sử dụng template (ví dụ: Bootstrap, TailwindCSS) để có giao diện sạch sẽ, tối giản mà không tốn thời gian code CSS. Logic JavaScript chỉ dùng để gọi API (fetch) và hiển thị dữ liệu (DOM manipulation).

3.6 DevOps & Chất lượng Mã nguồn (QA)

- **Quản lý Mã nguồn:** Git, GitHub
- **Công cụ Chất lượng:** pre-commit (bắt buộc).
- **Formatter:** black (Tự động format code).
- **Linters:** flake8 (Kiểm tra lỗi logic và tuân thủ style guide PEP 8).

Phần 4: Mô hình Vận hành & Triết lý Nhóm

4.1 Triết lý Cốt lõi: "Sở hữu Chung, Chuyên môn Hóa Linh hoạt"

Đây là một dự án "Python-centric", do đó, mọi thành viên trong nhóm đều là "Python Developer" trước tiên, bất kể vai trò "Lead" được giao. Chúng ta áp dụng triết lý "Sở hữu Chung" (Shared Ownership). Bất kỳ ai cũng có thể (và được khuyến khích) đóng góp vào bất kỳ phần nào của mã nguồn (đặc biệt là 5 module Python backend).

Chúng ta chống lại "Silo Kiến thức" (Knowledge Silos). Sẽ không có tình trạng "chỉ một người biết về NLP" hay "chỉ một người biết về API". Tất cả các module phức tạp (đặc biệt là Module 4 và 5) được khuyến khích thực hiện thông qua lập trình cặp (pair programming) hoặc lập trình nhóm (mob programming).

4.2 Cấu trúc Vai trò "Lead" (Vai trò Mũ, không phải Chức danh)

Để đảm bảo trật tự và trách nhiệm, chúng ta phân công 4 vai trò "Lead". "Lead" không có nghĩa là "chỉ làm việc đó", mà là người chịu trách nhiệm cuối cùng về chiến lược kỹ thuật, chất lượng và định hướng của mảng đó. Trách nhiệm chính của "Lead" là hướng dẫn kỹ thuật, đảm bảo các module tích hợp được với nhau, và review Pull Request (PR) liên quan đến mảng của mình. Tuy nhiên, phần lớn thời gian code của tất cả thành viên (kể cả Frontend Lead) phải được dành cho việc xây dựng các module Python backend.

4.2.1 Backend Lead

- **Trách nhiệm:** Chịu trách nhiệm cuối cùng về thiết kế CSDL (SQLite), kiến trúc API (Flask), và đảm bảo pipeline xử lý dữ liệu (từ API đến các module Python) được tích hợp mượt mà.

4.2.2 NLP Lead

- **Trách nhiệm:** Chịu trách nhiệm cuối cùng về nghiên cứu, thiết kế và chất lượng/độ chính xác của các Module 1, 2, 3, 5, và 6 (File Processing, NER, Tóm tắt, Matching, Deduplication).

4.2.3 Frontend Lead

- **Trách nhiệm:** Chịu trách nhiệm cuối cùng về việc đảm bảo giao diện Public UI và Admin UI được giữ "tối giản", có thể gọi API thành công và hiển thị kết quả. Ngăn chặn mọi nỗ lực "phức tạp hóa" giao diện. Bắt buộc phải tham gia code Module 4 (Scoring) và Module 5 (Matching) cùng nhóm.

4.2.4 DevOps/QA Lead

- **Trách nhiệm:** Chịu trách nhiệm cuối cùng về việc thiết lập và duy trì repository (Git). Cấu hình và thực thi `pre-commit` (black, flake8). Quản lý việc triển khai (deploy) MVP lên nền tảng miễn phí. Viết các Unit Test cơ bản cho các module Python.

Phần 5: Quy trình Kiểm soát Chất lượng & Định nghĩa Hoàn thành (DoD)

5.1 Quy trình Phát triển (Git Workflow)

- **Quy trình:** Áp dụng "Feature Branch Workflow" (Quy trình Nhánh Tính năng).
 1. Tất cả mã nguồn mới (tính năng, sửa lỗi) phải được phát triển trên một nhánh riêng (feature branch), ví dụ: `feature/module-2-ner`.
 2. Nhánh `main` được bảo vệ (protected) và không ai được phép push trực tiếp.
 3. Mã nguồn chỉ được hợp nhất (merge) vào `main` thông qua "Yêu cầu Kéo" (Pull Request - PR).
- **Yêu cầu Pull Request (PR) Bắt buộc:**
 1. Mỗi PR phải được review và "Approved" (Chấp thuận) bởi ít nhất 1 thành viên khác (ưu tiên "Lead" của mảng đó).
 2. PR phải vượt qua tất cả các kiểm tra tự động của `pre-commit`.

5.2 Đảm bảo Chất lượng Tự động (QA)

Việc cài đặt và sử dụng `pre-commit` là bắt buộc với mọi thành viên. Một hook `pre-commit` sẽ được cấu hình trong repository để tự động chạy `black` (để format code) và `flake8` (để kiểm

tra lỗi và tuân thủ PEP 8) trước khi một commit được phép tạo. Cam kết (commit) sẽ tự động bị từ chối nếu `black` hoặc `flake8` báo lỗi. Điều này đảm bảo 100% mã nguồn trên nhánh `main` luôn sạch sẽ, nhất quán và tuân thủ tiêu chuẩn chất lượng.

5.3 Định nghĩa Hoàn thành (Definition of Done - DoD)

Đây là các tiêu chí mang tính ràng buộc để xác định khi nào công việc thực sự "hoàn thành".

5.3.1 DoD cho một Feature (hoặc Module Python)

- Mã nguồn đã được merge vào nhánh `main`.
- Mã nguồn (Python) vượt qua 100% kiểm tra của `black` và `flake8`.
- Mã nguồn có tài liệu (docstrings) đầy đủ cho các hàm/class chính.
- (Nếu có thể) Có Unit Test cơ bản cho logic nghiệp vụ quan trọng.

5.3.2 DoD cho MVP (Toàn dự án)

1. MVP (với 5 module Python đầu tiên) được triển khai thành công lên một nền tảng công khai (ví dụ: Render, PythonAnywhere, Hugging Face Spaces) và có thể truy cập được qua URL.
2. Hệ thống hoạt động end-to-end: (1) HR tạo Job -> (2) Ứng viên upload CV -> (3) HR thấy CV đã được xử lý (trích xuất, tóm tắt, chấm điểm, so khớp) trong dashboard.
3. 100% mã nguồn Python trong repository tuân thủ `black` và `flake8`.
4. Tài liệu `README.md` hoàn chỉnh, hướng dẫn rõ ràng cách cài đặt, chạy dự án local, và cung cấp liên kết (link) đến bản demo đã triển khai.
5. (Quan trọng) Mỗi thành viên trong nhóm đã đóng góp (authored) và merge thành công ít nhất 2 Pull Request chứa logic Python nghiệp vụ quan trọng (không tính PR sửa lỗi nhỏ, tài liệu, hay code UI).

5.3.3 DoD cho Mục tiêu Mở rộng (Stretch Goal)

- Trình bày được khả năng "hoán đổi" (swap) linh hoạt giữa hai phương pháp tóm tắt (NLTK và AI Transformers) thông qua một cờ (flag) cấu hình hoặc một API endpoint khác.
- Trình bày được tính năng phát hiện trùng lặp `fuzzywuzzy` trong Admin UI.

Phần 6: Lộ trình Tích hợp và Phân rã Công việc (12 Tuần)

Lộ trình chiến lược này được thiết kế để hoàn thành và đóng băng tính năng MVP vào cuối Tuần 10, dành 2 tuần cuối cho việc tinh chỉnh, triển khai, và nghiên cứu các mục tiêu mở rộng. Rủi ro lớn nhất là "Địa ngục Tích hợp" (Integration Hell), do đó Giai đoạn 2 tập trung vào việc tích hợp các module vào pipeline một cách tuần tự.

GIẢI ĐOẠN 1: NỀN TẢNG & MODULE CỐT LÕI (Tuần 1-4)

Tuần 1-3: Khởi động & Xây dựng Module Độc lập

- **Nhiệm vụ Backend Lead (Lead: A):**
 - Thiết kế CSDL (SQLite) & Models.
 - Tạo API `POST /job`, `GET /jobs` (chưa có NLP).
 - Hoàn thiện Models (Candidate, CV_File).
 - Viết API "câm" `POST /apply` (chỉ nhận file, lưu file, tạo record CSDL).
 - Tinh chỉnh API.
- **Nhiệm vụ NLP Lead (Lead: B):**
 - Nghiên cứu `PyPDF2` & `python-docx`.
 - Viết **Module 1 (Xử lý File)** (dưới dạng script độc lập).
 - Viết **Module 2 (NER)** (dưới dạng script độc lập, nhận text, trả về JSON/dict).
 - Viết **Module 3 (Tóm tắt NLTK)** (dưới dạng script độc lập, tuân thủ interface `BaseSummarizer`).
- **Nhiệm vụ Frontend Lead (Lead: C):**
 - Tìm kiếm template.
 - Dựng UI tĩnh (HTML/CSS) cho trang Public (danh sách job).
 - Dựng UI tĩnh (HTML/CSS) cho trang Admin (dashboard).
 - Dựng UI tĩnh cho form `Apply` (upload).
 - Dùng JavaScript (fetch) để gọi API `GET /jobs` và hiển thị Job lên trang Public.
- **Nhiệm vụ DevOps/QA Lead (Lead: D):**
 - Thiết lập Repo (Git), nhánh `main`.
 - Cấu hình `pre-commit` (`black`, `flake8`) và yêu cầu cả nhóm cài đặt.
 - Viết Unit Test cơ bản cho Module 1.
 - Hỗ trợ nhóm về Git, PR.
 - Viết Unit Test cơ bản cho Module 2.

Tuần 4: Hoàn thiện Module Giai đoạn 1

- **Nhiệm vụ Backend Lead (Lead: A):**
 - Tích hợp **Module 1** vào API `/apply` (bóc tách text và lưu `raw_text` vào CSDL).
 - Hỗ trợ tích hợp Module 2.
 - Tạo API `GET /admin/candidates`.
- **Nhiệm vụ NLP Lead (Lead: B):**
 - Hoàn thiện Module 1, 2, 3.

- Tích hợp **Module 2** vào API `/apply` (cùng Backend Lead, lưu thông tin NER vào CSDL).
 - **Nhiệm vụ Frontend Lead (Lead: C):**
 - Dùng JavaScript để gọi API `POST /apply` (upload file thành công).
 - Xây dựng trang Admin UI, gọi `GET /admin/candidates`.
 - **Nhiệm vụ DevOps/QA Lead (Lead: D):**
 - Đảm bảo 100% commit tuân thủ `flake8`.
 - Review PR của các thành viên.
 - Bắt đầu nghiên cứu nền tảng triển khai (Render/PythonAnywhere).
-

GIẢI ĐOẠN 2: TÍCH HỢP PIPELINE (Tuần 5-8)

Tuần 5: Tích hợp Pipeline (End-to-End)

- **Nhiệm vụ Backend Lead (Lead: A):**
 - Tích hợp **Module 3 (NLTK)** vào API `GET /job/<id>` và `GET /candidate/<id>`.
- **Nhiệm vụ NLP Lead (Lead: B):**
 - Hỗ trợ Backend Lead tích hợp Module 3.
 - Bắt đầu nghiên cứu **Module 5 (scikit-learn)**.
- **Nhiệm vụ Frontend Lead (Lead: C):**
 - Hiển thị Tóm tắt (NLTK) của Job trên trang Public.
 - Hiển thị thông tin (NER) và Tóm tắt CV trên trang Admin.
- **Nhiệm vụ DevOps/QA Lead (Lead: D):**
 - Viết Test Tích hợp cho luồng `/apply` (từ upload đến khi thấy dữ liệu NER trong CSDL).

Tuần 6: Bắt đầu Module ML

- **Nhiệm vụ Backend Lead (Lead: A):**
 - Hỗ trợ nhóm thiết kế Module 4.
 - Tạo API để HR định nghĩa "trọng số" kỹ năng cho Job.
- **Nhiệm vụ NLP Lead (Lead: B):**
 - Nghiên cứu và xây dựng **Module 5 (Matching: TF-IDF & Cosine Similarity)**.
- **Nhiệm vụ Frontend Lead (Lead: C):**
 - Tham gia code (pair programming) **Module 4 (Scoring)** cùng nhóm (theo triết lý "Sở hữu chung").
- **Nhiệm vụ DevOps/QA Lead (Lead: D):**

- Review PR.
- Đảm bảo chất lượng code Module 4.

Tuần 7: Tích hợp Module ML

- **Nhiệm vụ Backend Lead (Lead: A):**
 - Tham gia code (pair programming) **Module 4 (Scoring)**.
 - Tích hợp Module 4 (Scoring) vào pipeline `/apply`.
- **Nhiệm vụ NLP Lead (Lead: B):**
 - Hoàn thiện và tinh chỉnh Module 5.
 - Hỗ trợ tích hợp Module 4 & 5.
- **Nhiệm vụ Frontend Lead (Lead: C):**
 - Hoàn thiện UI Admin để hiển thị điểm số từ Module 4.
- **Nhiệm vụ DevOps/QA Lead (Lead: D):**
 - Viết Unit Test cho Module 4 (logic chấm điểm).

Tuần 8: Tích hợp Module ML (tiếp)

- **Nhiệm vụ Backend Lead (Lead: A):**
 - Tích hợp **Module 5 (Matching)** vào pipeline `/apply`.
 - **Nhiệm vụ NLP Lead (Lead: B):**
 - Hỗ trợ tích hợp Module 5.
 - Đánh giá độ chính xác của các module.
 - **Nhiệm vụ Frontend Lead (Lead: C):**
 - Hiển thị điểm số (Module 4) và điểm so khớp (Module 5) trên Admin UI.
 - **Nhiệm vụ DevOps/QA Lead (Lead: D):**
 - Rà soát toàn bộ mã nguồn.
 - Chuẩn bị môi trường triển khai (Staging).
-

GIẢI ĐOẠN 3: HOÀN THIỆN MVP & NÂNG CAO (Tuần 9-12)

Tuần 9: Hoàn thiện MVP

- **Cả nhóm (Pair/Mob Programming):**
 - Tinh chỉnh, sửa lỗi, và tối ưu hóa 5 module Python.
 - Đảm bảo pipeline hoạt động mượt mà.
 - Frontend Lead hoàn thiện UI cuối cùng.
 - DevOps Lead hoàn thiện Unit Test.

Tuần 10: Đóng băng Tính năng MVP & Triển khai

- **Nhiệm vụ Backend Lead (Lead: A):**
 - Hỗ trợ triển khai.
 - Bắt đầu viết tài liệu `README.md`.
- **Nhiệm vụ NLP Lead (Lead: B):**
 - Đánh giá cuối cùng về độ chính xác của MVP.
 - Hỗ trợ viết tài liệu.
- **Nhiệm vụ Frontend Lead (Lead: C):**
 - Kiểm thử giao diện lần cuối.
 - Quay video demo MVP.
- **Nhiệm vụ DevOps/QA Lead (Lead: D):**
 - **Triển khai (Deploy) MVP** lên nền tảng miễn phí.
 - Hoàn thiện `README.md` (hướng dẫn cài đặt).

Tuần 11: Nghiên cứu Mục tiêu Mở rộng (Stretch)

- **Nhiệm vụ Backend Lead (Lead: A):**
 - (Nếu MVP ổn định) Bắt đầu nghiên cứu **Nâng cấp Module 3 (AI Transformers)**.
- **Nhiệm vụ NLP Lead (Lead: B):**
 - (Nếu MVP ổn định) Bắt đầu nghiên cứu **Nâng cấp Module 3 (AI Transformers)** và **Module 6 (fuzzywuzzy)**.
- **Nhiệm vụ Frontend Lead (Lead: C):**
 - Hỗ trợ demo và tinh chỉnh UI (nếu cần).
- **Nhiệm vụ DevOps/QA Lead (Lead: D):**
 - Đảm bảo bản triển khai (production) ổn định.
 - Chuẩn bị slide demo cuối kỳ.

Tuần 12: Demo & Báo cáo Cuối kỳ

- **Nhiệm vụ Backend Lead (Lead: A):**
 - Chuẩn bị demo (phần Backend/API).
 - Hoàn thiện báo cáo.
- **Nhiệm vụ NLP Lead (Lead: B):**
 - Chuẩn bị demo (phần NLP/ML).
 - (Nếu kịp) Trình bày khả năng "hoán đổi" (swap) NLTK/AI.
- **Nhiệm vụ Frontend Lead (Lead: C):**
 - Chuẩn bị demo (phần Frontend/Luồng người dùng).
- **Nhiệm vụ DevOps/QA Lead (Lead: D):**

- Trình bày về Quy trình QA/DevOps và demo bản triển khai (production).
- Merge PR cuối cùng.