

ĐƠN VỊ QUẢN LÝ: PHÒNG BIG DATA

Đề tài 1: Xây dựng hệ thống RAG (Retrieval-Augmented Generation) hỗ trợ hỏi đáp tự động

1. Mục tiêu đề tài

Phát triển một hệ thống hỏi đáp thông minh sử dụng mô hình RAG, kết hợp truy xuất dữ liệu và mô hình ngôn ngữ lớn (LLM) để sinh câu trả lời chính xác và tự nhiên từ tài liệu nội bộ của doanh nghiệp.

2. Yêu cầu đầu vào

- Bộ tài liệu nội bộ: PDF, DOCX, HTML hoặc Markdown (vd: hướng dẫn sử dụng hệ thống, tài liệu kỹ thuật, tài liệu nghiệp vụ...).
- Tập câu hỏi mẫu để đánh giá hiệu quả hệ thống.
- Môi trường triển khai: máy chủ ảo/PC cá nhân (có thể cài Docker, Python).

3. Yêu cầu đầu ra

- Hệ thống hỏi đáp đầu cuối, gồm các thành phần:
 - o Xử lý tài liệu và sinh embedding;
 - o Vector database (vd: sử dụng FAISS hoặc Weaviate);
 - o Tích hợp mô hình LLM (vd: OpenAI GPT-4, Mistral, Claude hoặc local LLM);
 - o API hoặc giao diện web để người dùng đặt câu hỏi và nhận phản hồi.
- Độ chính xác trả lời đạt từ 70% trở lên trên bộ câu hỏi kiểm thử.
- Hệ thống có khả năng mở rộng dữ liệu đầu vào.

4. Tài liệu kỹ thuật cần nộp

- **Báo cáo kỹ thuật (Technical Report):**
 - o Tổng quan mô hình RAG;
 - o Kiến trúc hệ thống;
 - o Mô tả chi tiết pipeline (xử lý dữ liệu, truy xuất, sinh văn bản);
 - o Đánh giá hiệu năng (precision, recall, latency).
- **Code đầy đủ** (trong repo GitHub hoặc nộp file ZIP).
- **Hướng dẫn cài đặt và chạy thử (README.md).**

Đề tài 2: Xây dựng hệ thống OCR thông minh cho tài liệu Tiếng Việt

1. Mục tiêu đề tài Phát triển hệ thống OCR (Optical Character Recognition) chuyên biệt cho văn bản Tiếng Việt, xử lý được các loại tài liệu: hóa đơn, chứng từ, form, tài liệu scan với độ chính xác cao.

2. Yêu cầu đầu vào

- Bộ dữ liệu hình ảnh tài liệu Tiếng Việt đa dạng (scan, photo, PDF)
- Ground truth text annotations
- Các loại font chữ và style khác nhau (in nghiêng, đậm, mờ)
- Môi trường triển khai: máy chủ ảo/PC cá nhân (có thể cài Docker, Python).

3. Yêu cầu đầu ra

- Hệ thống OCR hoàn chỉnh:
 - Phát hiện văn bản
 - Nhận diện văn bản Tiếng Việt (vd: PaddleOCR/TrOCR, opencv)
 - Tiền xử lý để cải thiện độ chính xác
 - JSON/XML output với độ tin cậy
 - Web interface để upload và xử lý tài liệu
- Độ chính xác ở mức ký tự $\geq 95\%$
- Hỗ trợ xử lý hàng loạt

4. Tài liệu kỹ thuật cần nộp

- **Báo cáo kỹ thuật:** Tổng quan OCR pipeline, thuật toán text detection, sequence-to-sequence models
- **Evaluation scripts và test results**
- **Training code và model checkpoints đầy đủ** (trong repo GitHub hoặc nộp file ZIP).
- **Hướng dẫn cài đặt và chạy thử (README.md).**

Đề tài 3: Xây dựng hệ thống giám sát bãi đỗ xe thông minh sử dụng thị giác máy tính

1. Mục tiêu đề tài

Phát triển một hệ thống giám sát bãi đỗ xe tự động sử dụng thị giác máy tính và trí tuệ nhân tạo để phát hiện và theo dõi các xe cộ, quản lý chỗ đỗ xe, nhận diện biển số xe và cung cấp thông tin thời gian thực về tình trạng bãi đỗ xe.

2. Yêu cầu đầu vào

- Video từ hệ thống camera giám sát bãi đỗ xe.
- Bộ dữ liệu hình ảnh xe cộ và biển số Việt Nam để huấn luyện mô hình.
- Tập dữ liệu kiểm thử với các tình huống khác nhau (ban ngày, ban đêm, thời tiết khác nhau).
- Môi trường triển khai: máy chủ ảo/PC cá nhân(có thể cài Docker, Python, CUDA).

3. Yêu cầu đầu ra

- Hệ thống giám sát bãi đỗ xe đầu cuối, gồm các thành phần:
 - Xử lý video thời gian thực và phát hiện phương tiện (vd: sử dụng YOLO, R-CNN, hoặc SSD);
 - Mô-đun nhận diện biển số xe với OCR cho ký tự Việt Nam;
 - Thuật toán theo dõi đối tượng đa mục tiêu (vd: DeepSORT, ByteTrack, hoặc FairMOT);
 - Cơ sở dữ liệu lưu trữ thông tin xe vào/ra, thời gian đỗ, vị trí đỗ;
 - API hoặc giao diện web/mobile hiển thị thông tin bãi đỗ xe và thống kê.
- Độ chính xác phát hiện xe $\geq 90\%$ và nhận diện biển số $\geq 85\%$ trên dữ liệu kiểm thử.
- Tốc độ xử lý: ≥ 15 FPS cho video stream thời gian thực.
- Hệ thống có khả năng mở rộng cho nhiều camera và bãi đỗ xe khác nhau.

4. Tài liệu kỹ thuật cần nộp

- **Báo cáo kỹ thuật (Technical Report):**
 - Tổng quan các phương pháp phát hiện và theo dõi phương tiện hiện tại;
 - Kiến trúc hệ thống và luồng xử lý dữ liệu;
 - Mô tả chi tiết pipeline (tiền xử lý video, phát hiện xe, nhận diện biển số, quản lý chỗ đỗ);

- Đánh giá hiệu năng (mAP cho detection, accuracy cho ANPR, processing time, throughput);
- Phân tích các thách thức (occlusion, lighting conditions, weather effects).
- **Code đầy đủ (trong repo GitHub hoặc nộp file ZIP) bao gồm mô hình đã huấn luyện và database schema.**
- **Hướng dẫn cài đặt và chạy thử (README.md) với demo ứng dụng.**

Đề tài 4: Xây dựng hệ thống chống giả mạo khuôn mặt (Anti-Spoofing) cho xác thực sinh trắc học

1. Mục tiêu đề tài

Phát triển hệ thống trí tuệ nhân tạo phát hiện và ngăn chặn các cuộc tấn công giả mạo khuôn mặt (sử dụng ảnh in, video phát lại, mặt nạ 3D).

2. Yêu cầu đầu vào

- Bộ dữ liệu khuôn mặt thật và giả mạo đa dạng:
 - Khuôn mặt thật từ camera trực tiếp
 - Tấn công bằng ảnh in (photo attack)
 - Tấn công phát lại video (replay attack)
- Video thu thập từ nhiều thiết bị khác nhau (smartphone, webcam, camera an ninh)
- Điều kiện ánh sáng và góc nhìn đa dạng
- Môi trường triển khai: máy chủ ảo/PC cá nhân (có thể cài Docker, Python, CUDA).

3. Yêu cầu đầu ra

- Hệ thống chống giả mạo khuôn mặt đầu cuối, gồm các thành phần:
 - Xử lý video/ảnh và trích xuất đặc trưng đa phương thức (RGB, depth, texture);
 - Mô hình học sâu phát hiện giả mạo (vd: sử dụng CNN, Vision Transformer, hoặc ResNet kết hợp với LSTM);
 - Mô-đun kiểm tra hoạt động sống (liveness detection) (vd: sử dụng màu sắc ánh sáng, cử động);
 - API hoặc giao diện web để kiểm tra ảnh/video và nhận kết quả phân loại thật/giả.
- Độ chính xác phát hiện giả mạo đạt từ 95% trở lên trên bộ dữ liệu kiểm thử.
- Tỷ lệ báo động nhầm (False Positive Rate) $\leq 2\%$.

- Thời gian xử lý $\leq 500\text{ms}$ cho mỗi frame video.

4. Yêu cầu đầu ra chi tiết

- **Báo cáo kỹ thuật (Technical Report):** Tổng quan các phương pháp chống giả mạo khuôn mặt hiện tại; Kiến trúc hệ thống và các thành phần chính; Mô tả chi tiết pipeline (tiền xử lý, trích xuất đặc trưng, phân loại, hậu xử lý); Đánh giá hiệu năng (APCER, BPCER, ACER, độ trễ xử lý); So sánh với các phương pháp baseline và các giải pháp thương mại.
- **Code đầy đủ (trong repo GitHub hoặc nộp file ZIP) bao gồm mô hình đã huấn luyện.**
- **Hướng dẫn cài đặt và chạy thử (README.md) với demo ứng dụng**

Đề tài 5: Xây dựng hệ thống quản lý video (VMS)

1. Mục tiêu đề tài

Phát triển một hệ thống cho phép giám sát và lưu dữ liệu từ camera.

2. Yêu cầu đầu vào

- Bộ tài liệu nội bộ: PDF, DOCX (vd: hướng dẫn sử dụng hệ thống, tài liệu kỹ thuật, tài liệu nghiệp vụ...).
- Môi trường triển khai: máy chủ ảo/PC cá nhân (có thể cài Docker,..).

3. Yêu cầu đầu ra

Hệ thống đáp ứng các chức năng sau:

* Quản lý camera:

- Thêm, cấu hình và nhóm nhiều camera IP từ các hãng khác nhau.
- Hỗ trợ ONVIF hoặc giao thức riêng.
- Điều khiển PTZ (Pan-Tilt-Zoom) nếu camera hỗ trợ.

* Xem lại video (Playback & Search)

- Tìm kiếm video theo ngày giờ, camera
- Giao diện timeline giúp tua nhanh, tua chậm.

* Lưu trữ & Quản lý dữ liệu

- Lưu video vào ổ cứng cục bộ
- Tùy chỉnh thời gian lưu trữ theo chính sách (ví dụ: 7 ngày, 30 ngày)
- Có thể xuất file video (định dạng MP4, AVI) khi cần bằng chứng

* Phân quyền người dùng

- Tạo nhiều tài khoản người dùng

- Phân quyền theo nhóm: xem được camera nào, được xem live hay playback, được xuất video không...

* Giao diện trực quan

- Hiện thị đồng thời nhiều camera dạng lưới

- Kéo thả, phóng to từng camera.

4. Tài liệu kỹ thuật cần nộp

- **Báo cáo kỹ thuật (Technical Report):**
 - o Tài liệu HLD
 - o Tài liệu HDSD
- **Code đầy đủ** (trong repo GitHub hoặc nộp file ZIP).
- **Hướng dẫn cài đặt và chạy thử (README.md).**

ĐƠN VỊ QUẢN LÝ: PHÒNG PHÁT TRIỂN PHẦN MỀM 2

Đề tài 6: Xây dựng hệ thống hội nghị truyền hình

1. Mục tiêu đề tài

Tham gia phát triển hệ thống hội nghị truyền hình (hỗ trợ đa điểm, chia sẻ màn hình, audio/video chất lượng ổn định). Thu thập yêu cầu, đánh giá hiệu năng, ghi nhận bài học để mở rộng/ứng dụng mô hình tại doanh nghiệp.

2. Yêu cầu đầu vào

- Có kiến thức về mạng, giao thức SIP/WebRTC.
- Biết Linux, Asterisk, Docker, lập trình scripting bash, Python, JS.
- Ưu tiên ứng viên đã từng làm project liên quan real-time communications.
- Kỹ năng tiếng Anh cơ bản (đọc hiểu Tài liệu kỹ thuật)

3. Yêu cầu đầu ra

- Hoàn thành xây dựng các module trong hệ thống hội nghị truyền hình theo yêu cầu được giao
- Viết được tài liệu hướng dẫn deploy hệ thống (kèm mã nguồn, phiên bản, lệnh cấu hình).

4. Tài liệu kỹ thuật cần nộp

- Thực hiện báo cáo nghiệm thu (có số liệu: băng thông, độ trễ, CPU/RAM).
- Tóm tắt các bài học (khó khăn, cải tiến, đề xuất)
- **Code đầy đủ** (trong repo GitHub hoặc nộp file ZIP).

Đề tài 7: Xây dựng hệ thống hỗ trợ quản lý thời gian làm việc và báo cáo công việc theo tuần

1. Mục tiêu đề tài

Xây dựng một hệ thống web giúp người dùng (nhân viên hoặc sinh viên thực tập) ghi nhận, quản lý và báo cáo công việc theo ngày và tuần. Hệ thống cho phép:

- Nhập liệu các đầu việc theo ngày, theo dõi tiến độ theo trạng thái.
- Tổng hợp, thống kê và xuất báo cáo tuần dạng PDF hoặc Excel.
- Hỗ trợ quản lý (leader) theo dõi hiệu quả làm việc của từng người trong nhóm.

Mục tiêu đề tài nhằm giúp sinh viên:

- Rèn luyện kỹ năng phân tích, thiết kế hệ thống thực tế.
- Áp dụng thành thạo các công nghệ phát triển phần mềm web.
- Hiểu rõ quy trình báo cáo – quản lý công việc trong môi trường doanh nghiệp.

2. Yêu cầu đầu vào

Sinh viên cần chuẩn bị/đáp ứng:

- Kiến thức lập trình web cơ bản (Java hoặc Python hoặc JavaScript).
- Biết sử dụng một framework backend (Spring Boot / Flask / Express...).
- Kiến thức cơ bản về CSDL quan hệ (PostgreSQL hoặc MySQL).
- Tư duy UI/UX cơ bản (giao diện người dùng trực quan, dễ sử dụng).
- Sử dụng Git để quản lý mã nguồn.

Dữ liệu đầu vào của hệ thống gồm:

- Danh sách người dùng (email, tên, phân quyền).
- Thông tin đầu việc: tiêu đề, mô tả, thời gian thực hiện, trạng thái (todo, doing, done), deadline.

3. Yêu cầu đầu ra

Hệ thống phần mềm có thể chạy độc lập và có giao diện web, gồm các tính năng:

a. Với nhân viên:

- Đăng nhập vào hệ thống.
- Thêm, chỉnh sửa, xóa các đầu việc theo ngày.
- Giao diện nhập công việc đơn giản, hỗ trợ tìm kiếm/lọc theo ngày/tuần.
- Theo dõi tiến độ (dựa trên trạng thái công việc).

b. Với trưởng nhóm:

- Xem báo cáo tổng hợp theo từng người, theo tuần.
- Tùy chọn xuất báo cáo (PDF hoặc Excel).
- Có thể lọc công việc theo người dùng, theo khoảng thời gian.

c. Kỹ thuật:

- Hệ thống sử dụng kiến trúc client-server (REST API).
- Giao diện web thân thiện, responsive.
- Có thể chạy bằng Docker (ưu tiên).
- Không yêu cầu triển khai thực tế trên server, chỉ cần demo được qua mạng nội bộ hoặc localhost.

4. Tài liệu kỹ thuật cần nộp

Sinh viên nộp các tài liệu sau vào cuối kỳ thực tập:

Báo cáo phân tích và thiết kế hệ thống:

- Sơ đồ chức năng (Use case diagram).
- Sơ đồ cơ sở dữ liệu.
- Mô tả kiến trúc hệ thống.

Hướng dẫn sử dụng hệ thống (User Manual):

- Cách cài đặt, chạy hệ thống.
- Cách sử dụng các chức năng.

Source code đầy đủ trên GitHub hoặc nộp file nén:

- Ghi rõ commit, hướng dẫn run bằng Docker hoặc chạy thủ công.

File báo cáo tuần:

- Tổng hợp những gì đã làm trong từng tuần thực tập (có thể dạng bảng đơn giản).

Đề tài 8: Ứng dụng liên lạc OTT - Gọi điện & Nhắn tin đa phương tiện

1. Mục tiêu đề tài:

Xây dựng một ứng dụng liên lạc OTT (Over-The-Top) hoạt động trên nền tảng mobile, cho phép người dùng:

- Gọi điện Internet (Voice/Video call) giữa hai người hoặc nhóm.
- Nhắn tin thời gian thực, hỗ trợ gửi văn bản, hình ảnh, video, file đính kèm.
- Quản lý danh bạ người dùng: thêm/xóa bạn bè, tìm kiếm và kết bạn.

- Hiện thị cuộc gọi đến dưới dạng màn hình đồ chuông (CallKit trên iOS và CallKeep trên Android).
- Xem lịch sử cuộc gọi và tin nhắn.
- Đăng nhập/đăng ký và quản lý tài khoản người dùng.

2. Yêu cầu đầu vào:

- Nắm vững kiến thức cơ bản về lập trình mobile (Android Java, IOS Swift, React Native hoặc Flutter).
- Kiến thức về mạng, WebSocket, REST API, và giao tiếp thời gian thực (WebRTC).
- Tư duy logic, kỹ năng debug tốt, có tinh thần học hỏi và chủ động nghiên cứu.
- Có thể đọc hiểu tài liệu kỹ thuật tiếng Anh.
- Kiến thức về quản lý trạng thái (Redux, Zustand, hoặc Context API), Firebase, và push notification.

3. Yêu cầu đầu ra:

- Ứng dụng OTT hoàn chỉnh, hỗ trợ đa nền tảng (iOS/Android) với các chức năng:
 - Gọi thoại/video peer-to-peer (WebRTC).
 - Nhắn tin real-time sử dụng Socket.io hoặc Firebase.
 - Gửi/nhận ảnh, video, tài liệu (media files), chưa yêu cầu các action nâng cao như pin message, trả lời, lưu trữ, reaction, remove message,...
 - Hiện thị thông báo khi có tin nhắn/cuộc gọi mới (Push Notification, VoIP Push).
 - Màn hình đồ chuông khi có cuộc gọi đến (CallKeep/CallKit).
 - Danh bạ bạn bè, trạng thái online/offline.
 - Đăng nhập, đăng ký, cập nhật hồ sơ người dùng.
- Hiệu năng tốt, giao diện trực quan và trải nghiệm người dùng mượt mà.

4. Tài liệu kỹ thuật cần nộp:

Báo cáo kỹ thuật (Technical Report):

- Kiến trúc hệ thống tổng thể (backend - frontend - signaling - push).
- Thiết kế luồng hoạt động chính (Use Case, Sequence Diagram).
- Tài liệu OOAD (Object-Oriented Analysis and Design).

Mã nguồn đầy đủ:

- Nộp dạng GitHub repo hoặc file ZIP.

Tài liệu triển khai (README.md):

- Hướng dẫn cài đặt và cấu hình môi trường (Android Studio, Xcode, Firebase, v.v.).
- Hướng dẫn chạy thử hệ thống trên thiết bị thật hoặc giả lập.
- Các yêu cầu đặc biệt (permissions, VoIP capability...).

Tài liệu sử dụng:

- I. Hướng dẫn đăng ký, gọi điện, nhắn tin, thêm bạn...
- II. Hướng dẫn thêm cấu hình Firebase, CallKit, Push Notification.

ĐƠN VỊ QUẢN LÝ: PHÒNG PHÁT TRIỂN PHẦN MỀM 1

Đề tài 9: Ứng dụng Quiz/Trivia đơn giản

1. Mục tiêu đề tài:

Xây dựng một ứng dụng đồ vui (quiz) với một tập hợp các câu hỏi và lựa chọn trả lời. Người dùng sẽ trả lời các câu hỏi và nhận được điểm số cuối cùng.

Hiển thị từng câu hỏi một.

Các lựa chọn trả lời (radio buttons hoặc checkboxes).

Kiểm tra đáp án và hiển thị kết quả đúng/sai ngay lập tức hoặc sau khi hoàn thành.

Tính điểm và hiển thị tổng điểm khi kết thúc quiz.

Có thể thêm chức năng reset quiz.

2. Yêu cầu đầu vào

- Có kiến thức cơ bản về lập trình: JavaScript, HTML, CSS
- Ham học hỏi và nghiên cứu
- Có sở thích làm việc với hệ thống, tìm hiểu kiến thức mới
- Kỹ năng đọc hiểu tài liệu tiếng anh

3. Yêu cầu đầu ra

- Kiến thức, kinh nghiệm có được từ dự án
- Hiểu và nắm rõ được các topic nghiên cứu
- Ứng dụng đồ vui hoàn chỉnh với chức năng đầy đủ, giao diện bắt mắt, trải nghiệm người dùng mượt mà, hiệu năng tốt, tương thích với nhiều loại thiết bị
- Dự án phát triển sử dụng TypeScript, React. Ngoài ra có thể sử dụng thêm: Tanstack Query, Tailwind, ...

4. Tài liệu kỹ thuật cần nộp

- **Báo cáo kỹ thuật (Technical Report):**
 - Kiến trúc hệ thống tổng thể.
 - Tài liệu OOAD
- **Mã nguồn đầy đủ (nộp dưới dạng repo Git hoặc file ZIP).**
- **Hướng dẫn triển khai: cài đặt, cấu hình, chạy thử hệ thống (README.md).**
- **Tài liệu sử dụng: hướng dẫn sinh câu hỏi, các dữ liệu liên quan và hướng dẫn sử dụng hệ thống**

Đề tài 10: Ứng dụng AI gợi ý lộ trình học tập cá nhân hóa

1. Mục tiêu đề tài: Nhập thông tin kỹ năng hiện tại + mục tiêu nghề nghiệp (VD: Backend Developer, Data Analyst...). Hệ thống dùng AI để gợi ý roadmap học tập cá nhân.

2. Yêu cầu đầu vào

- Thông tin đầu vào từ người dùng:
 - Danh sách kỹ năng hiện có và mức độ thành thạo (ví dụ: Python - Trung bình, HTML - Cơ bản).
 - Mục tiêu nghề nghiệp mong muốn (chọn từ danh sách có sẵn hoặc nhập tự do).
 - Thời gian học trung bình mỗi tuần (giờ/tuần).
 - Thời hạn hoàn thành mong muốn (ví dụ: 6 tháng, 1 năm).
- Dữ liệu tham chiếu:
 - Bộ kỹ năng cần thiết cho từng nghề nghiệp phổ biến (do hệ thống thu thập và cập nhật từ các nguồn uy tín như Coursera, Job Description, LinkedIn...).
 - Tập tài nguyên học tập được gắn thẻ kỹ năng (khóa học, bài viết, dự án mẫu).
- Môi trường triển khai: máy chủ/PC có thể chạy Docker, Python và tích hợp API AI.

3. Yêu cầu đầu ra

- Hệ thống tư vấn học tập đầu cuối, gồm các thành phần:

- Phân tích chênh lệch kỹ năng giữa hiện tại và mục tiêu.
- Gợi ý roadmap học tập theo tuần/tháng với trình tự hợp lý.
- Gợi ý tài nguyên học tập cụ thể cho từng kỹ năng (link khóa học, bài thực hành, dự án mẫu...).
- API hoặc giao diện web để người dùng nhập thông tin và nhận gợi ý trực tiếp.
- Hệ thống có khả năng:
 - Tùy biến roadmap theo mục tiêu khác nhau và thời lượng học tập cụ thể.
 - Điều chỉnh lộ trình nếu người dùng cập nhật kỹ năng hoặc thay đổi mục tiêu.

4. Tài liệu kỹ thuật cần nộp

- **Báo cáo kỹ thuật (Technical Report):**
 - Mô tả bài toán và phương pháp đề xuất.
 - Kiến trúc hệ thống tổng thể.
 - Chi tiết pipeline AI: phân tích kỹ năng, lập kế hoạch học tập, lựa chọn tài nguyên.
 - Đánh giá chất lượng gợi ý (theo user feedback hoặc scoring từ chuyên gia).
- **Mã nguồn đầy đủ (nộp dưới dạng repo GitHub hoặc file ZIP).**
- **Hướng dẫn triển khai: cài đặt, cấu hình, chạy thử hệ thống (README.md).**
- **Tài liệu sử dụng: hướng dẫn người dùng nhập liệu và đọc kết quả gợi ý.**

Đề tài 11: Trợ lý cá nhân hóa cho chuyến đi du lịch

1. Mục tiêu đề tài

Xây dựng hệ thống gợi ý thông tin để đi du lịch dựa trên những input được cấp (địa điểm, chi phí, mong muốn về địa điểm thuê khách sạn, ...). Hệ thống dùng AI để gợi ý chuyến đi du lịch.

2. Yêu cầu đầu vào

Để xây dựng hệ thống, thực tập sinh sẽ làm việc với các loại dữ liệu sau:

- **Tài liệu tham khảo:** Các bộ tài liệu liên quan đến du lịch (địa điểm, khách sạn, hoạt động), có thể ở định dạng PDF, DOCX, HTML, Markdown.
- **Tập câu hỏi mẫu:** Để kiểm thử và đánh giá hiệu quả của hệ thống.
- **Môi trường làm việc:** Hệ thống sẽ được phát triển và kiểm thử trên môi trường có thể cài đặt **Docker** và **Python** hoặc **Java Development Kit (JDK)** (ví dụ: máy chủ ảo, PC cá nhân).

3. Yêu cầu đầu ra

Kết quả của dự án này sẽ bao gồm:

- **Hệ thống trợ lý du lịch hoàn chỉnh:**
 - **Module xử lý dữ liệu:** Xử lý và chuyển đổi tài liệu thô thành định dạng phù hợp.
 - **Cơ sở dữ liệu:** Lưu trữ và truy vấn thông tin hiệu quả (có thể là cơ sở dữ liệu truyền thống hoặc vector database).
 - **Tích hợp công cụ AI:** Sử dụng các công cụ AI có sẵn hoặc API để xử lý thông tin và đưa ra gợi ý thông minh.
 - **Giao diện người dùng:** API hoặc giao diện web (UI) thân thiện để người dùng tương tác, nhập yêu cầu và nhận gợi ý.
- **Độ chính xác:** Hệ thống cần đạt độ chính xác từ **70%** trở lên trong việc đưa ra gợi ý phù hợp dựa trên bộ câu hỏi kiểm thử.
- **Khả năng mở rộng:** Hệ thống phải có khả năng dễ dàng mở rộng dữ liệu đầu vào và các tính năng trong tương lai.

4. Tài liệu kỹ thuật cần nộp

Để đánh giá kết quả và quy trình làm việc, thực tập sinh cần chuẩn bị:

- **Báo cáo kỹ thuật (Technical Report):**
 - **Tổng quan:** Giới thiệu về hệ thống và mục tiêu.
 - **Kiến trúc:** Mô tả chi tiết cấu trúc tổng thể của hệ thống.
 - **Quy trình:** Giải thích luồng dữ liệu và xử lý trong hệ thống (từ đầu vào đến đầu ra).

- **Đánh giá hiệu năng:** Các chỉ số đánh giá chất lượng hệ thống (ví dụ: độ chính xác của gợi ý, thời gian phản hồi).
- **Mã nguồn (Code):** Toàn bộ mã nguồn dự án được tổ chức rõ ràng trong kho GitHub hoặc dưới dạng file ZIP.
- **Hướng dẫn cài đặt & chạy thử (README.md):** Tài liệu hướng dẫn chi tiết các bước để cài đặt và chạy thử hệ thống.