

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



CÔNG NGHỆ PHẦN MỀM (CO3001)

Báo cáo bài tập lớn

BKArch: Hệ thống hỗ trợ chương trình Tutor/Mentor

Giảng viên hướng dẫn: Mai Đức Trung

STT	Họ tên SV	MSSV	Lớp
1	Nguyễn Đức Nghĩa	2312266	L03
2	Phan Huy Quang Minh	2312105	L03
3	Ngô Thanh Hùng	2311289	L03
4	Nguyễn Đắc Nghĩa	2312265	L03
5	Hà Cao Đức Minh	2312059	L03
6	Nguyễn Trí Hải	2314047	L02
7	Trần Tiến Khải	2311566	L03

Thành phố Hồ Chí Minh, tháng 10 năm 2025



Nội dung

Danh sách thành viên và khối lượng công việc	2
1 Tổng quan về dự án	3
1.1 Bối cảnh và tính cấp thiết	3
1.2 Mục đích và mục tiêu dự án	3
1.2.1 Mục đích dự án:	3
1.2.2 Mục tiêu dự án:	4
1.3 Định hướng kỹ thuật	4
1.3.1 Mô hình Server-Client	4
1.3.2 Flask Framework	5
1.3.3 MongoDB	6
1.3.4 Git và GitHub	7
1.4 Định hướng mở rộng và nâng cấp	7
1.4.1 Tích hợp AI	7
1.4.2 Xây dựng các tính năng cộng đồng và các diễn đàn:	7
1.4.3 Chương trình học thuật và phi học thuật:	8
1.4.4 Cá nhân hoá học tập:	8
1.5 Phạm vi dự án	8
1.5.1 Chức năng cốt lõi	8
1.5.2 Chức năng mở rộng (đề xuất)	9
2 Yêu cầu phi chức năng (Non-Functional Requirements)	9
2.1 Hiệu năng (Performance)	9
2.2 Bảo mật (Security)	10
2.3 Khả năng mở rộng (Scalability)	10
2.4 Khả năng tích hợp (Interoperability)	10
2.5 Tính khả dụng (Availability)	10
2.6 Tính dễ sử dụng (Usability)	10
2.7 Khả năng giám sát và bảo trì (Maintainability & Monitoring)	10



Danh sách thành viên và khối lượng công việc

STT	Họ và tên	MSSV	Vấn đề	% hoàn thành
1	Nguyễn Đức Nghĩa	2312266	A	100%
2	Phan Huy Quang Minh	2312105	B	100%
3	Ngô Thanh Hùng	2311289	C	100%
4	Nguyễn Đắc Nghĩa	2312265	D	100%
5	Hà Cao Đức Minh	2312059	E	100%
6	Nguyễn Trí Hải	2314047	F	100%
7	Trần Tiến Khải	2311566	G	100%

1 Tổng quan về dự án

1.1 Bối cảnh và tính cấp thiết

Hiện nay, tại Trường Đại học Bách Khoa – ĐHQG TP.HCM, chương trình Tutor/Mentor đã được triển khai nhằm hỗ trợ sinh viên trong quá trình học tập và phát triển kỹ năng, thông qua việc kết nối họ với các giảng viên, nghiên cứu sinh hoặc sinh viên có học lực tốt. Tuy nhiên, công tác quản lý chương trình vẫn đang thực hiện chủ yếu bằng phương pháp thủ công, rời rạc và thiếu một nền tảng thống nhất, gây khó khăn trong việc theo dõi, sắp xếp lịch gặp gỡ, cũng như đánh giá hiệu quả hỗ trợ. Trong bối cảnh số lượng sinh viên tham gia ngày càng tăng, nhu cầu xây dựng một hệ thống số hiện đại, đồng bộ và dễ mở rộng trở nên cấp thiết hơn bao giờ hết. Từ thực tế đó, chúng tôi đã phát triển ứng dụng BKArch với mục tiêu số hóa toàn bộ quy trình của chương trình Tutor/Mentor. Ứng dụng này sẽ giúp đơn giản hóa việc đăng ký, phân công, đặt lịch và quản lý buổi gặp, đồng thời hỗ trợ các bên liên quan dễ dàng theo dõi, đánh giá và khai thác dữ liệu để nâng cao chất lượng học tập cũng như tối ưu hóa nguồn lực của nhà trường.

1.2 Mục đích và mục tiêu dự án

1.2.1 Mục đích dự án:

Ứng dụng hỗ trợ Tutor được xây dựng nhằm:

- Cung cấp nền tảng hỗ trợ học tập và phát triển kỹ năng cho sinh viên một cách có hệ thống.
- Quản lý và vận hành chương trình Tutor/Mentor một cách hiện đại, hiệu quả và có khả năng mở rộng trong tương lai.
- Tăng cường kết nối giữa sinh viên và Mentor nhằm nâng cao chất lượng học tập và giảng dạy.
- Tích hợp chặt chẽ với hạ tầng công nghệ hiện có của nhà trường nhằm đảm bảo tính bảo mật, nhất quán dữ liệu nội bộ, đồng thời tạo nền tảng cho việc mở rộng và phát triển hệ sinh thái số hỗ trợ học tập, kết nối và quản lý toàn diện trong môi trường đại học



1.2.2 Mục tiêu dự án:

Ứng dụng được xây dựng nhằm đáp ứng các mục tiêu cụ thể sau đây:

Nhóm mục tiêu	Mô tả chi tiết
Quản lý dữ liệu người dùng	<ul style="list-style-type: none">– Lưu trữ thông tin cá nhân của sinh viên và tutor (chuyên môn, nhu cầu hỗ trợ, thời gian sẵn sàng).– Cho phép sinh viên đăng ký chương trình và được phân công tutor (thủ công hoặc tự động).
Tổ chức, quản lý buổi tư vấn	<ul style="list-style-type: none">– Thiết lập, hủy hoặc thay đổi lịch hẹn.– Hỗ trợ hình thức gặp trực tiếp và trực tuyến.– Tích hợp thông báo tự động, nhắc lịch và ghi nhận buổi gặp (nếu cần).
Đánh giá và phản hồi	<ul style="list-style-type: none">– Sinh viên đánh giá chất lượng buổi tư vấn.– Tutor theo dõi tiến trình học tập.– Khoa/Bộ môn phân tích dữ liệu để theo dõi kết quả học tập.– Phòng CTSV sử dụng dữ liệu để xét điểm rèn luyện hoặc học bổng.
Tích hợp hệ thống của trường	<ul style="list-style-type: none">– Kết nối hệ thống đăng nhập tập trung HCMUT_SSO.– Đồng bộ dữ liệu từ HCMUT_DATACORE.– Phân quyền truy cập theo vai trò.– Liên kết HCMUT_LIBRARY để chia sẻ tài liệu học tập.
Tính năng nâng cao	<ul style="list-style-type: none">– Ghép cặp thông minh bằng AI.– Cộng đồng trực tuyến cho tutor và mentee.– Hỗ trợ các chương trình học thuật và phi học thuật.– Cá nhân hóa hỗ trợ học tập bằng AI.

1.3 Định hướng kỹ thuật

1.3.1 Mô hình Server-Client

Mô hình Server-Client (Máy chủ - Máy khách) là một mô hình kiến trúc trong đó các chức năng được phân chia rõ ràng:

- **Server (Máy chủ):** Đóng vai trò là một nhà cung cấp dịch vụ (service provider). Nó là một máy tính (hoặc hệ thống phần mềm) mạnh mẽ, luôn trong trạng thái chờ đợi và lắng nghe các yêu cầu từ client. Khi nhận được yêu cầu, server sẽ xử lý và cung cấp tài nguyên, dữ liệu hoặc dịch vụ phù hợp trả về cho client.
- **Client (Máy khách):** Đóng vai trò là người yêu cầu dịch vụ (service requester). Nó là một thiết bị (máy tính, điện thoại, trình duyệt web, ứng dụng) gửi yêu cầu đến server để sử dụng tài nguyên hoặc dịch vụ mà server cung cấp.

Việc lựa chọn kiến trúc máy khách-máy chủ thay vì mạng ngang hàng (P2P) thường mang lại lợi thế khi các yêu cầu cụ thể về kiểm soát tập trung, bảo mật, khả năng mở rộng và tính tin cậy là yếu tố tối quan trọng.

Lý do nên chọn mô hình Máy khách-Máy chủ:

1. **Quản lý và kiểm soát tập trung:** Mô hình máy khách-máy chủ cho phép một cơ quan có thẩm quyền tập trung (máy chủ) quản lý tài nguyên, dữ liệu và quyền truy cập của người dùng. Điều này giúp đơn giản hóa công tác quản trị, cập nhật và bảo trì hệ thống tổng thể.
2. **Bảo mật nâng cao:** Với một máy chủ tập trung, các biện pháp bảo mật có thể được triển khai và thực thi một cách nhất quán trên tất cả các máy khách. Dữ liệu có thể được lưu trữ và bảo vệ trong một môi trường được kiểm soát chặt chẽ, đồng thời các quyền truy cập có thể được quản lý một cách chính xác. Ngược lại, các mạng P2P có thể dễ bị tổn thương hơn trước các vi phạm bảo mật vì mỗi peer (nút ngang hàng) đều là một điểm tấn công tiềm năng.
3. **Khả năng mở rộng:** Kiến trúc máy khách-máy chủ nhìn chung có khả năng mở rộng tốt hơn, cho phép bổ sung thêm nhiều máy khách và tài nguyên mà không ảnh hưởng đáng kể đến hiệu suất. Các máy chủ chuyên dụng có thể được nâng cấp và tối ưu hóa để xử lý khối lượng công việc ngày càng tăng.
4. **Độ tin cậy và Khả năng sẵn sàng cao:** Một máy chủ được bảo trì tốt có thể mang lại thời gian hoạt động liên tục (uptime) và độ tin cậy cao hơn so với mạng P2P, nơi khả năng sẵn có của tài nguyên phụ thuộc vào việc từng peer riêng lẻ có đang trực tuyến và hoạt động hay không.
5. **Tính nhất quán và toàn vẹn dữ liệu:** Việc lưu trữ dữ liệu tập trung trên máy chủ đảm bảo tính nhất quán và toàn vẹn của thông tin trên tất cả các máy khách, ngăn ngừa các xung đột hoặc sai lệch tiềm ẩn có thể phát sinh trong môi trường P2P phi tập trung.
6. **Hiệu suất tốt hơn cho các ứng dụng cụ thể:** Đối với các ứng dụng đòi hỏi xử lý hiệu năng cao, lưu trữ dữ liệu lớn hoặc các thao tác phức tạp, một máy chủ chuyên dụng có thể cung cấp các tài nguyên và sức mạnh xử lý cần thiết một cách hiệu quả hơn so với các peer riêng lẻ.
7. **Khắc phục sự cố dễ dàng hơn:** Với một điểm kiểm soát tập trung, việc xác định và giải quyết các sự cố trong mạng máy khách-máy chủ thường đơn giản và trực quan hơn so với việc chẩn đoán vấn đề trên vô số các peer riêng lẻ trong một hệ thống P2P.

1.3.2 Flask Framework

Flask là một micro-framework web nhẹ và mạnh mẽ được viết bằng Python. Về cơ bản, Flask cung cấp những công cụ tối thiểu để xây dựng một ứng dụng web, từ đó lập trình viên có thể tự do lựa chọn các công nghệ và thư viện khác đi kèm.

Trong project này, nhóm sử dụng Flask cho cả **Frontend** và **Backend**:

- **Backend (API & Logic):** Flask xử lý các yêu cầu HTTP, nghiệp vụ logic, xử lý dữ liệu và kết nối cơ sở dữ liệu.
- **Frontend (Giao diện người dùng):** Flask sử dụng Jinja2 (một template engine mạnh mẽ) để render các file HTML, kết hợp với CSS và JavaScript nhằm tạo thành giao diện hoàn chỉnh mà người dùng nhìn thấy. Về bản chất, nhóm đang áp dụng mô hình *Server-Side Rendering (SSR)*.

Lợi ích khi sử dụng Flask cho dự án:

1. **Nhẹ và tối giản:** Flask là một microframework, nghĩa là nó chỉ cung cấp các chức năng cốt lõi mà không áp đặt các cấu trúc cứng nhắc hay thành phần không cần thiết. Điều này cho phép nhà phát triển xây dựng ứng dụng chỉ với những tính năng cần thiết, giúp mã nguồn gọn nhẹ và hiệu quả hơn.
2. **Linh hoạt và có khả năng tùy chỉnh cao:** Nhờ tính tối giản, Flask mang lại sự linh hoạt tối đa. Nhà phát triển có toàn quyền kiểm soát kiến trúc ứng dụng và có thể tích hợp các thư viện/công cụ ưa thích mà không bị ràng buộc bởi framework.
3. **Khả năng mở rộng tốt:** Thiết kế module của Flask cho phép tích hợp dễ dàng với nhiều tiện ích mở rộng và thư viện, giúp nhà phát triển có thể thêm các tính năng như xác thực người dùng, kết nối cơ sở dữ liệu hoặc kiểm tra dữ liệu biểu mẫu khi cần.
4. **Gỡ lỗi nhanh chóng:** Flask tích hợp sẵn máy chủ phát triển (development server) và trình gỡ lỗi (debugger), giúp tối ưu hóa quá trình phát triển và khắc phục sự cố một cách nhanh chóng.
5. **Phù hợp cho dự án nhỏ đến trung bình và API:** Flask đặc biệt phù hợp để xây dựng ứng dụng web quy mô nhỏ, API và microservice nhờ sự nhẹ nhàng và linh hoạt. Tính đơn giản của Flask cũng giúp tạo ra các nguyên mẫu và sản phẩm khả thi tối thiểu (MVP) hiệu quả.

1.3.3 MongoDB

MongoDB là một cơ sở dữ liệu NoSQL phổ biến theo hướng tài liệu (*document-oriented*), lưu trữ dữ liệu trong các tài liệu linh hoạt, có dạng giống JSON thay vì các rigid table.

Các ưu điểm chính của MongoDB bao gồm:

- **Tính linh hoạt và lược đồ mềm dẻo:** Lược đồ động cho phép nhà phát triển nhanh chóng điều chỉnh mô hình dữ liệu theo các yêu cầu ứng dụng luôn thay đổi, giúp nó phù hợp với các ứng dụng hiện đại có cấu trúc dữ liệu biến đổi linh hoạt.
- **Khả năng mở rộng và phân mảnh dữ liệu (Sharding):** MongoDB được thiết kế để mở rộng theo chiều ngang, cho phép xử lý các tập dữ liệu lớn bằng cách phân phối dữ liệu trên nhiều máy chủ (sharding).
- **Tính sẵn sàng cao:** Các tính năng như nhân bản (replication) cho phép chuyển đổi dự phòng tự động, đảm bảo dữ liệu luôn khả dụng và hệ thống có khả năng phục hồi ngay cả khi một máy chủ gặp sự cố.
- **Hiệu suất cao:** Bằng cách lưu trữ dữ liệu trong RAM và cung cấp khả năng lập chỉ mục (indexing) và truy vấn mạnh mẽ, MongoDB mang lại hiệu suất cao cho việc truy xuất và xử lý dữ liệu.
- **Dễ sử dụng:** Mô hình dữ liệu dạng văn bản (document) của nó thường ánh xạ trực quan hơn với các ngôn ngữ lập trình hướng đối tượng, giúp đơn giản hóa quá trình phát triển và tăng tốc độ xử lý.
- **Hỗ trợ đa dạng kiểu dữ liệu:** MongoDB quản lý hiệu quả khối lượng lớn dữ liệu có cấu trúc, bán cấu trúc và không có cấu trúc.



1.3.4 Git và GitHub

Git là hệ thống quản lý phiên bản phân tán, giúp ghi lại mọi thay đổi trong mã nguồn. GitHub là dịch vụ lưu trữ trên web dành cho các dự án sử dụng Git, cung cấp các công cụ để cộng tác.

1.3.4.1 Điểm cộng khi dùng Git/GitHub cho một dự án app/web:

- **Quản lý phiên bản an toàn:** Theo dõi mọi thay đổi code, dễ dàng khôi phục lại trạng thái ổn định trước đó nếu xảy ra lỗi.
- **Hợp tác nhóm hiệu quả:** Nhiều người có thể làm việc song song trên các tính năng khác nhau và hợp nhất lại một cách dễ dàng, tránh xung đột.
- **Theo dõi công việc minh bạch:** Sử dụng Issues và Pull Requests để phân công task, thảo luận và kiểm tra code trước khi đưa vào dự án chính.
- **Triển khai tự động (CI/CD):** Tích hợp với các công cụ để tự động kiểm thử, build và triển khai ứng dụng lên môi trường production.

1.4 Định hướng mở rộng và nâng cấp

Hệ thống có thể được mở rộng và tích hợp thêm các tính năng sau: tích hợp trí thông minh nhân tạo (AI), xây dựng các tính năng cộng đồng, trang diễn đàn cho sinh viên, hỗ trợ thực hiện các chương trình học thuật và phi học thuật.

1.4.1 Tích hợp AI

AI được tích hợp vào để đánh giá, phân tích tình hình học tập của sinh viên cũng như hỗ trợ sinh viên trong học tập.

- **Phân tích, đánh giá tình hình học tập của sinh viên:** Tổng hợp lại điểm số của sinh viên qua từng kỳ, đánh giá tình hình hiện tại của sinh viên, đưa ra các cảnh báo hoặc lưu ý khi thấy điểm số có biến động.
- **Hỗ trợ sinh viên trong học tập:** Dựa vào số liệu tổng hợp được và nhu cầu của từng sinh viên, đưa ra các lộ trình học tập phù hợp với từng cá nhân. Sử dụng các kỹ thuật AI để tối ưu hóa việc ghép đôi giữa giảng viên và sinh viên dựa trên các tiêu chí cụ thể như lĩnh vực chuyên môn của giảng viên và nhu cầu hỗ trợ của sinh viên.

1.4.2 Xây dựng các tính năng cộng đồng và các diễn đàn:

Cung cấp các tính năng cộng đồng và các diễn đàn trực tuyến, nơi sinh viên và giảng viên có thể kết nối, thảo luận và chia sẻ kiến thức, tạo ra một môi trường học tập và trao đổi mở rộng bên ngoài các buổi học trên lớp.

Diễn đàn thảo luận:

- **Tạo bài viết và chủ đề:** Giảng viên và sinh viên có thể đăng các câu hỏi, chia sẻ tài liệu hữu ích, và bắt đầu các cuộc thảo luận về các môn học hoặc lĩnh vực chuyên môn.
- **Phân loại chủ đề:** Chủ đề có thể được phân loại theo môn học, ngành học hoặc các thể (tag) liên quan để dễ dàng tìm kiếm và theo dõi.
- **Bình luận và tương tác:** Người dùng có thể bình luận, trả lời, và tương tác với bài viết của người dùng khác, trao đổi kiến thức và kinh nghiệm.



Tính năng cộng đồng:

- **Nhóm học tập:** Cho phép người dùng tạo các nhóm học tập nhỏ để cùng nhau làm bài tập hoặc ôn thi.
- **Tin nhắn riêng tư:** Các sinh viên, giảng viên có thể gửi tin nhắn cho nhau, có thể đánh dấu tin nhắn quan trọng.
- **Thông báo và cập nhật:** Thông báo cho người dùng về các tin nhắn riêng tư, các tin nhắn cộng đồng cũng như các bình luận trong bài viết mà người dùng đang theo dõi trong diễn đàn. Những tin nhắn quan trọng sẽ được làm nổi bật lên.

1.4.3 Chương trình học thuật và phi học thuật:

Hệ thống hỗ trợ các chương trình học thuật và phi học thuật do nhà trường hoặc bên thứ ba thông qua nhà trường tổ chức.

- **Tổ chức chương trình:** Những chương trình tổ chức sẽ được tổng hợp lại trong một trang riêng. Phân loại ra các hoạt động đã tổ chức và đang tổ chức. Mỗi hoạt động sẽ có thông tin chi tiết và cách tham dự.
- **Hỗ trợ sinh viên:** Hệ thống sẽ lưu trữ lại và hiển thị ra những chương trình mà sinh viên đã tham dự.

1.4.4 Cá nhân hoá học tập:

Hệ thống áp dụng AI để đưa ra phương pháp học tập, tài liệu học tập phù hợp với nhu cầu và mục tiêu của mỗi sinh viên.

- **Phân tích tình hình học tập:** Theo dõi tiến độ học tập và xác định các môn sinh viên đang gặp khó khăn.
- **Đề xuất phương pháp học tập phù hợp:** Đề xuất các tài liệu, bài giảng, bài tập hoặc các buổi tư vấn chuyên sâu phù hợp với điểm yếu của từng sinh viên. Sinh viên có thể lựa chọn phương pháp học tập phù hợp như tự học, học nhóm hoặc tư vấn 1-1 với giảng viên.

1.5 Phạm vi dự án

1.5.1 Chức năng cốt lõi

1. Quản lý người dùng & xác thực

- Đăng nhập tập trung thông qua **HCMUT_SSO**.
- Đồng bộ dữ liệu cá nhân từ **HCMUT_DATACORE**: họ tên, MSSV/Mã cán bộ, email học vụ, khoa/ngành, trạng thái học tập/giảng dạy.
- Quản lý phân quyền theo vai trò (sinh viên, tutor, điều phối viên, bộ môn, quản trị hệ thống).

2. Quản lý tutor & mentee

- Tutor quản lý hồ sơ chuyên môn, thời gian rảnh, danh sách sinh viên đang hướng dẫn.
- Sinh viên đăng ký tham gia chương trình, chọn hoặc được phân công tutor.
- Điều phối viên xem, phân công hoặc phê duyệt ghép cặp tutor – sinh viên.



3. Quản lý lịch & buổi tư vấn

- Tutor tạo lịch rảnh và mở buổi tư vấn (offline hoặc online).
- Sinh viên đặt lịch gặp tutor, với hỗ trợ hủy/đổi/dời lịch.
- Gửi thông báo tự động (email, notification) khi có thay đổi lịch.
- Tự động nhắc lịch (reminder) trước buổi gặp.
- Cho phép ghi nhận nội dung và biên bản buổi gặp (nếu cần).

4. Đánh giá & phản hồi

- Sinh viên phản hồi, đánh giá chất lượng buổi học/tutor.
- Tutor theo dõi tiến bộ và ghi nhận kết quả học tập của mentee.
- Bộ môn và Phòng Đào tạo khai thác dữ liệu để tổng hợp tình hình học tập.

5. Quản lý học liệu & tài nguyên

- Tích hợp với **HCMUT_LIBRARY** để truy cập, chia sẻ tài liệu, giáo trình và sách.

1.5.2 Chức năng mở rộng (đề xuất)

1. Hệ thống AI ghép cặp và tương tác trực tuyến

- Ứng dụng AI để gợi ý phân công tutor – mentee tối ưu, dựa trên hồ sơ cá nhân, chuyên môn và nhu cầu hỗ trợ.
- Tạo cộng đồng trực tuyến cho tutor – mentee nhằm trao đổi, thảo luận và chia sẻ kinh nghiệm.
- Tích hợp chat nhóm/cá nhân, cho phép giao tiếp nhanh và nhận thông báo, nhắc nhở lịch hẹn ngay trong hệ thống.

2. Hỗ trợ học tập cá nhân hóa bằng AI

- AI phân tích tiến trình học tập, dữ liệu phản hồi và kết quả rèn luyện.
- Đưa ra gợi ý học tập, tài liệu hoặc phương pháp phù hợp với từng sinh viên.

3. Quản lý Seminar/Event

- Cho phép tạo và quản lý sự kiện học thuật hoặc kỹ năng mềm.
- Sinh viên đăng ký tham gia, check-in bằng QR hoặc thẻ sinh viên.
- Hệ thống quản lý số lượng tham dự và phản hồi sau sự kiện.

2 Yêu cầu phi chức năng (Non-Functional Requirements)

2.1 Hiệu năng (Performance)

Hệ thống phải xử lý yêu cầu đăng ký, đặt lịch và truy xuất thông tin trong vòng dưới 2 giây. Hệ thống có khả năng phục vụ đồng thời tối thiểu 500 người dùng mà không bị gián đoạn. Thời gian phản hồi trung bình cho các thao tác thường xuyên không vượt quá 1 giây.



2.2 Bảo mật (Security)

Tích hợp xác thực tập trung HCMUT_SSO để đảm bảo đăng nhập an toàn và thống nhất. Dữ liệu cá nhân phải được mã hóa khi lưu trữ và truyền tải. Phân quyền truy cập theo vai trò được đồng bộ từ hệ thống tập trung. Ghi nhận nhật ký truy cập và thao tác để phục vụ kiểm tra bảo mật.

2.3 Khả năng mở rộng (Scalability)

Hệ thống phải có khả năng mở rộng để phục vụ số lượng người dùng tăng theo từng học kỳ. Kiến trúc hệ thống phải hỗ trợ bổ sung các chức năng nâng cao như AI ghép cặp hoặc cộng đồng trực tuyến.

2.4 Khả năng tích hợp (Interoperability)

Tích hợp với HCMUT_DATACORE để đồng bộ dữ liệu cá nhân và trạng thái học tập. Kết nối với HCMUT_LIBRARY để chia sẻ tài liệu học tập. Hỗ trợ API để tích hợp với các hệ thống nội bộ khác.

2.5 Tính khả dụng (Availability)

Hệ thống phải hoạt động liên tục 24/7 với thời gian uptime tối thiểu 99.5% mỗi tháng. Có cơ chế sao lưu và phục hồi dữ liệu định kỳ.

2.6 Tính dễ sử dụng (Usability)

Giao diện thân thiện, hỗ trợ đa thiết bị (máy tính, điện thoại). Có hướng dẫn sử dụng rõ ràng cho từng nhóm người dùng. Hệ thống hỗ trợ tiếng Việt và tiếng Anh.

2.7 Khả năng giám sát và bảo trì (Maintainability & Monitoring)

Cho phép theo dõi tiến độ học tập, phản hồi chất lượng buổi học, và tổng hợp báo cáo cho các phòng ban. Hệ thống dễ bảo trì, cập nhật và nâng cấp.