**ĐẠI HỌC DUY TÂN**

**TRƯỜNG KHMT**

**KHOA CÔNG NGHỆ THÔNG TIN**

**--------------🙖🙐✰🙖🙐-------------**

****

**Tên đề tài:**

**XÂY DỰNG HỆ THỐNG QUẢN LÍ TRUNG TÂM DẠY HỌC CỜ VUA TÍCH HỢP AI HỖ TRỢ**

**HỌC VIÊN**

**(PRODUCT BACKLOG DOCUMENT)**

**GVHD:** Th.S Trần Huệ Chi

**Thành Viên**

Nguyễn Hoàng Anh - 27211221512

Cù Thị Ngọc Anh - 27201200079

Nguyễn Văn Đại - 27211244010

Nguyễn Gia Khang - 27211221506

Nguyễn Bá Phát - 27211201137

**Đà Nẵng, Tháng 03 năm 2025**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **THÔNG TIN DỰ ÁN** | | | | | | | | |
| **Dự án viết tắt** | | | **KING-CHESS** | | | | | |
| **Tên dự án** | | | Xây dựng hệ thống quản lý trung tâm dạy học cờ vua tích hợp AI hỗ trợ học viên | | | | | |
| **Thời gian bắt đầu** | | | 01/03/2025 | **Ngày kết thúc** | | | 20/05/2025 | |
| **Lead Institution** | | | Khoa Công Nghệ Thông Tin, Trường Đại Học Duy Tân | | | | | |
| **Giáo viên hướng dẫn** | | | ThS. Trần Huệ Chi  Email:  [tranhuechidt@gmail.com](mailto:%20tranhuechidt@gmail.com)  Phone: 0983751077 | | | | | |
| **Chủ sở hữu dự án & Chi tiết liên hệ** | | | Nguyễn Văn Đại  Email: [dai293720@gmail.com](mailto:dai293720@gmail.com)  Phone: 0798350734 | | | | | |
| **Đối tác** | | | Duy Tan University | | | | | |
| **Quản lý dự án & Scrum Master** | | | Nguyễn Văn Đại | [dai293720@gmail.com](mailto:dai293720@gmail.com) | | | | 0798350734 |
| **Thành viên nhóm** | | | Nguyễn Hoàng Anh | [hoanganhtt11@gmail.com](mailto:nguyenhnhatquang@gmail.com) | | | | 0823031503 |
| Cù Thị Ngọc Anh | cuthingocanh3006@gmail.com | | | | 0981039623 |
| Nguyễn Gia Khang | Kangyenzt203@gmail.com | | | | 0971080103 |
| Nguyễn Bá Phát | [nbp10122003@gmail.com](mailto:nbp10122003@gmail.com) | | | | 0934439281 |
| `**TÊN TÀI LIỆU** | | | | | | | | |
| **Tiêu đề tài liệu** | Product Backlog | | | | | | | |
| **Tác giả** | Nguyễn Hoàng Anh | | | | | | | |
| **Chức năng** | Thành viên | | | | | | | |
| **Ngày** | 09/03/2025 | | | **Tên tệp:** | 04.Product-Backlog\_Nhom50.docx | | | |
| **URL** |  | | | | | | | |
| **Access** | Khoa CNTT | | | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **LỊCH SỬ BẢN SỬA** | | | |
| **Version** | **Person** | **Date** | **Description** |
| 1.0 | Nguyễn Hoàng Anh | 09/03/2025 | Tạo tài liệu |
| 1.1 | Nguyễn Hoàng Anh | 28/03/2025 | Chỉnh sửa |

|  |  |  |  |
| --- | --- | --- | --- |
| **PHÊ DUYỆT TÀI LIỆU**  Cần có các chữ ký sau để phê duyệt tài liệu này | | | |
| **Người hướng dẫn** | | Trần Huệ Chi | **Signature:** |  |
| **Date:** | …./…./2025 |
| **Chủ sở hữu** | | Nguyễn Văn Đại | **Signature:** |  |
| **Date:** | …./…./2025 |
| **Scrum Master** | | Nguyễn Văn Đại | **Signature:** |  |
| **Date:** | …./…./2025 |
| **Thành viên** | | Nguyễn Hoàng Anh | **Signature:** |  |
| **Date:** | …./…./2025 |
| Cù Thị Ngọc Anh | **Signature:** |  |
| **Date:** | …./…./2025 |
| Nguyễn Gia Khang | **Signature:** |  |
| **Date:** | …./…./2025 |
| Nguyễn Bá Phát | **Signature:** |  |
| **Date:** | 0…./…./2025 |

**DANH MỤC**

**[1. GIỚI THIỆU 1](#_Toc32022)**

[1.1. Mục đích 1](#_Toc9589)

[1.2. Phạm vi 1](#_Toc32377)

[1.3. Tham khảo 2](#_Toc3957)

[1.4. Biểu đồ UseCase Tổng quát 3](#_Toc4370)

**[2. PRODUCT BACKLOG 4](#_Toc5242)**

[2.1. Danh sách các Product Backlog trong Hệ thống 4](#_Toc16312)

[2.1.1 PB01-Đăng ký 7](#_Toc29451)

[2.1.2 PB02-Đăng nhập 8](#_Toc31133)

*[2.1.3 PB03-Đăng xuất](#_Toc1334)* [9](#_Toc1334)

*[2.1.4 PB04-Quên mật khẩu](#_Toc16818)* [9](#_Toc16818)

*[2.1.5 PB05-Tìm kiếm khóa học](#_Toc30423)* [11](#_Toc30423)

*[2.1.6 PB06-Quản lý thông tin cá nhân](#_Toc15945)* [12](#_Toc15945)

*[2.1.7 PB07-Quản lý khóa học](#_Toc32573)* [13](#_Toc32573)

*[2.1.8 PB08-Xem lịch học](#_Toc21173)* [14](#_Toc21173)

*[2.1.9 PB09-Xem lịch dạy](#_Toc16576)* [15](#_Toc16576)

*[2.1.10 PB10-Quản lý giỏ hàng](#_Toc31477)* [16](#_Toc31477)

*[2.1.11 PB11-Quản lý thanh toán](#_Toc17992)* [16](#_Toc17992)

*[2.1.12 PB12-Quản lý tài khoản](#_Toc8621)* [17](#_Toc8621)

*[2.1.13 PB13-Quản lý lớp học](#_Toc5591)* [18](#_Toc5591)

*[2.1.14 PB14-Quản lý bài thi](#_Toc11004)* [19](#_Toc11004)

[2.2. Phân tích về Sprint Backlog 20](#_Toc18578)

**[3. HẠN CHẾ 20](#_Toc754)**

**[4. TÓM LƯỢC 20](#_Toc23516)**

# 

# GIỚI THIỆU

## Mục đích

Mục đích của tài liệu này :

- Xây dựng một nền tảng trực tuyến giúp trung tâm dễ dàng quản lý học viên, giảng viên, khóa học và lịch học.

- Tích hợp trí tuệ nhân tạo (AI) để hỗ trợ tư vấn khách hàng, giúp học viên và phụ huynh nhanh chóng tìm kiếm thông tin mà không cần đến sự can thiệp của nhân viên.

- Áp dụng cổng thanh toán VNPAY, tạo ra quy trình giao dịch nhanh chóng, bảo mật và tiện lợi.

- Tăng cường trải nghiệm người dùng, giúp trung tâm vận hành hiệu quả hơn, nâng cao chất lượng dịch vụ và thu hút nhiều học viên hơn.

## Phạm vi

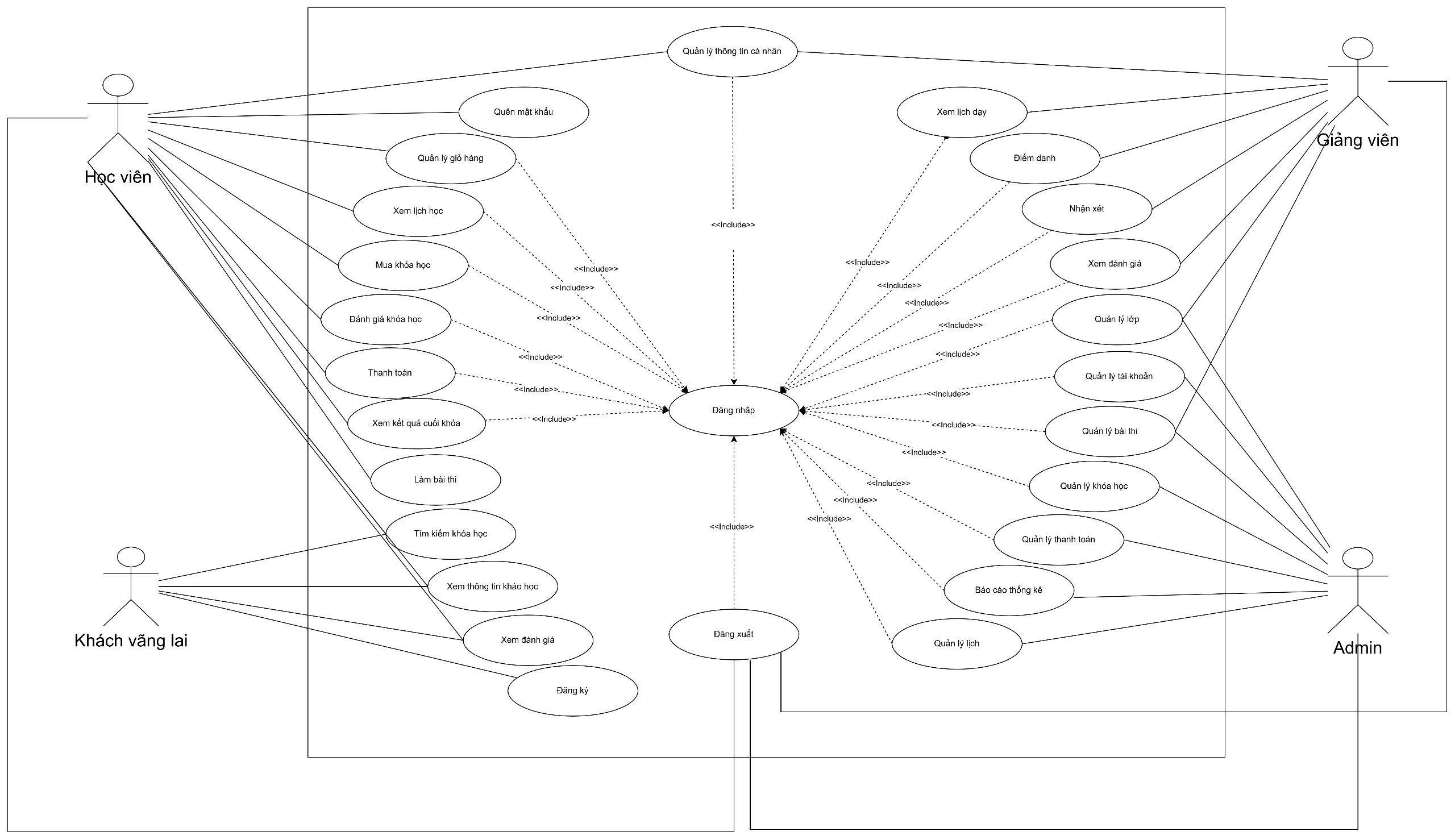
Đề tài tập trung vào việc xây dựng một hệ thống phần mềm quản lý trung tâm dạy học cờ vua tích hợp AI tư vấn khách hàng, nhằm số hóa quy trình quản lý vận hành và nâng cao trải nghiệm người dùng thông qua công nghệ. Phạm vi thực hiện Hệ thống sẽ bao gồm các chức năng chính sau: Quản lý khóa học: Thêm, sửa, xóa, phân loại khóa học theo trình độ và nội dung giảng dạy. Quản lý lớp học: Theo dõi sĩ số, phân công giáo viên, lịch học, trạng thái lớp. Quản lý học viên: Lưu trữ thông tin cá nhân, lịch sử học tập, kết quả học tập. Quản lý giáo viên: Thông tin, lịch dạy, lớp phụ trách và đánh giá chất lượng giảng dạy. AI tư vấn khách hàng: Gợi ý khóa học phù hợp dựa trên độ tuổi, trình độ, mục tiêu và thời gian rảnh của học viên. Quản lý thời khóa biểu: Tự động tạo và hiển thị thời khóa biểu cho giáo viên và học viên. Thanh toán học phí online qua VNPay: Tích hợp cổng thanh toán giúp học viên/phụ huynh thanh toán học phí trực tuyến an toàn, nhanh chóng. Báo cáo thống kê: Cung cấp số liệu theo từng lớp, khóa học, học viên hoặc thời gian.

## Tham khảo

*Table 1.1: Danh mục tài liệu tham khảo*

|  |  |  |
| --- | --- | --- |
| No | Document Information | References |
| 1 | Scrum Process | <https://www.mountaingoatsoftware.com/agile/scrum> |
| 2 | HTML, CSS, JS | https://www.w3schools.com/ |
| 3 | React | https://www.react.dev/ |
| 4 | NodeJS | https://www.nodejs.org/en/learn/getting-started/introduction-to-nodejs |
| 5 | Mongodb | https://www.mongodb.com/docs/ |
| 5 | Information | Requirement Document |

## Biểu đồ UseCase Tổng quát



# PRODUCT BACKLOG

2. 1. Danh sách các Product Backlog trong Hệ thống

*Bảng 1: Đặc tả tổng quát các Product Backlog*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Chủ đề** | **Với** | **Mong muốn** | **Nên** | **Độ ưu tiên** |
| PB01 | Đăng nhập | Học viên/Giảng viên/Admin | Truy cập vào hệ thống học tập | Người dùng có thể đăng nhập vào hệ thống | 1 |
| PB02 | Đăng ký | Khách vãng lai | Tạo tài khoản | Khách hàng có thể đăng ký tài khoản | 1 |
| PB03 | Đăng xuất | Học viên/Giảng viên/Admin | Thoát khỏi hệ thống | Người dùng có thể đăng xuất khỏi hệ thống | 2 |
| PB04 | Quên mật khẩu | Học viên/Giảng viên/Admin | Lấy lại mật khẩu khi quên | Người dùng có thể khôi phục mật khẩu của mình | 2 |
| PB05 | Tìm kiếm khóa học | Khách vãng lai/Học viên | Tìm khóa học phù hợp | Người dùng có thể tìm khóa học theo tên, cấp độ, giảng viên,… | 1 |
| PB06 | Quản lý thông tin cá nhân | Học viên | Cập nhật và quản lý thông tin cá nhân | Học viên có thể xem và chỉnh sửa thông tin của mình | 1 |
| PB07 | Quản lý khóa học | Admin | Quản lý nội dung khóa học | Có thể thêm, sửa, xóa khóa học | 1 |
| PB08 | Xem lịch học | Học viên | Xem lịch học theo từng buổi | Học viên có thể xem thời gian, địa điểm lớp học | 1 |
| PB09 | Xem lịch dạy | Giảng viên | Xem lịch giảng dạy | Giáo viên có thể xem lịch các lớp mình giảng dạy | 1 |
| PB10 | Quản lý Giỏ hàng | Học viên | Quản lý khóa học đã chọn | Người dùng có thể xem, chỉnh sửa các khóa học trong giỏ hàng | 1 |
| PB11 | Thanh toán | Học viên | Hoàn tất giao dịch mua khóa học | Người dùng có thể thanh toán khóa học một cách an toàn | 1 |
| PB12 | Quản lý tài khoản | Admin | Thêm/sửa/xóa tài khoản người dùng | Admin có thể quản lý tài khoản hệ thống | 1 |
| PB13 | Quản lý lớp học | Admin | Quản lý danh sách lớp học | Có thể thêm, chỉnh sửa, xóa lớp học | 1 |
| PB14 | Quản lý bài thi | Admin/ | Quản lý đề thi, điểm và lịch thi | Có thể thêm, sửa, phân phối bài thi | 2 |
| PB15 | Xem đánh giá | Khách vãng lai/Học viên | Xem nhận xét của học viên khác | Người dùng có thể xem đánh giá khóa học | 2 |
| PB16 | Mua khóa học | Học viên | Mua và đăng ký học | Học viên có thể chọn lịch, đăng ký và thanh toán khóa học | 1 |
| PB17 | Xem hóa đơn | Học viên | Theo dõi lịch sử mua hàng | Học viên có thể xem chi tiết hóa đơn đã thanh toán | 3 |
| PB18 | Đánh giá khóa học | Học viên | Đưa ra đánh giá về khóa học đã học | Học viên có thể đánh giá khóa học sau khi hoàn thành | 2 |
| PB19 | Điểm danh | Giảng viên | Ghi nhận tình trạng tham gia của học viên | Giáo viên có thể điểm danh theo buổi học | 1 |
| PB20 | Quản lý hóa đơn | Admin | Quản lý tất cả hóa đơn thanh toán của người dùng | Có thể tìm kiếm, xác nhận, và xuất hóa đơn | 2 |
| PB21 | Quản lý đánh giá | Admin | Theo dõi và kiểm duyệt đánh giá khóa học | Có thể xóa, chỉnh sửa hoặc duyệt đánh giá của học viên | 2 |
| PB22 | Thống kê | Admin | Thống kê số lượng học viên, doanh thu, tiến độ học | Có thể xuất báo cáo, biểu đồ phân tích dữ liệu | 2 |
| PB23 | Quản lý lịch | Admin | Quản lý toàn bộ lịch dạy và học của hệ thống | Có thể tạo, sửa, xóa lịch học/dạy của từng lớp/giáo viên/học viên | 1 |
| PB24 | Làm bài thi | Học viên | Làm bài thi , kiểm tra | Học viên làm bài thi | 1 |
| PB25 | AI tư vấn | Học viên | AI sẽ tư vấn học viên một cách chính xác nhất |  | 1 |

***\*Giải thích: Độ ưu tiên càng tăng thì càng quan trọng***

### 2.1.1 PB01-Đăng ký

|  |  |
| --- | --- |
| **ID** | **PB01** |
| **Tiêu đề** | Đăng ký |
| **Mô tả** | Khách vãng lai vào website để tạo tài khoản cá nhân |
| **Các bước thực hiện** | * Thiết kế giao diện đăng ký tài khoản. * Xử lý logic kiểm tra email đăng ký và gọi API đăng ký tài khoản. * Kiểm duyệt thông tin vừa đăng ký.   - Hiển thị thống báo và chuyển sang giao diện đăng nhập nếu thành công hoặc thông báo nếu đăng ký thất bại. |
| **Điều kiện trước** | - Khách hàng vào website  - Khách hàng bấm vào nút *đăng ký*  - Website sẽ đưa khách hàng đến trang đăng ký tài khoản  - Khách hàng nhập thông tin vào form đăng ký ( Email, password , confirm password , OTP )  - Kiểm tra nhập liệu (Yêu cầu khoảng trắng, số lượng ký tự, ký tự đặc biệt, v.v…)  + Nếu dữ liệu nhập vào thỏa mãn thì thông báo “Đăng ký thành công”  + Ngược lại, thống báo “Bạn nhập sai định dạng vui lòng nhập lại”  - Kiểm tra tồn tại. Nếu email đã tồn tại thì thông báo “Tên tài khoản này đã tồn tại”  - Chuyển hướng tới giao diện đăng nhập |
| **Điều kiện sau** | * Giao diện được tải thành công và khách vãng lai có thể thấy các nút tùy chọn chức năng. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện đăng kí) * **Backend:** Nodejs (xử lý xác thực) * **Database:** Mongodb(lưu trữ tài khoản) * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Cao |

### 2.1.2 PB02-Đăng nhập

|  |  |
| --- | --- |
| **ID** | **PB02** |
| **Tiêu đề** | Đăng nhập |
| **Mô tả** | Người dùng vào website, đã có tài khoản và đăng nhập vào hệ thống |
| **Các bước thực hiện** | * Thiết kế giao diện đăng nhập tài khoản. * Xử lý logic kiểm tra email đăng nhập và gọi API đăng nhập tài khoản. * Xử lý phân quyền sau khi đăng nhập.   - Hiển thị thống báo và chuyển sang giao diện trang chủ nếu thành công hoặc thông báo nếu đăng nhập thất bại. |
| **Điều kiện trước** | - Người dùng vào website  - Bấm vào nút *đăng nhập*  - Hệ thống hiển thị form đăng nhâp  + Người dùng nhập email, password  + Gửi yêu cầu đăng nhập đến hệ thống  - Kiểm tra tài khoản  + Kiểm tra sự tồn tại của email hoặc password.  + Nếu email hoặc password không tồn tại hiển thị thông báo “Email hoặc Password không đúng”  - Ngược lại, chuyển đến trang chủ của hệ thống |
| **Điều kiện sau** | * Giao diện được tải thành công và người dùng có thể thấy các nút tùy chọn chức năng. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện đăng nhập) * **Backend:** Nodejs (xử lý logic) * **Database:** Mongodb(lấy tài khoản) * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Cao |

### ***2.1.3 PB03-Đăng xuất***

|  |  |
| --- | --- |
| **ID** | **PB03** |
| **Tiêu đề** | Đăng xuất |
| **Mô tả** | Người dùng đăng xuất khỏi hệ thống và xóa thông tin xác thực khỏi phiên làm việc. |
| **Các bước thực hiện** | - Nhấn nút **“Đăng xuất”**  - Xóa **token đăng nhập** khỏi localStorage/cookie  - (Tùy chọn) Gọi API để đăng xuất phía backend  - Chuyển về **trang đăng nhập** hoặc **trang chủ**  - Hiển thị thông báo: **"Đăng xuất thành công"** |
| **Điều kiện trước** | - Người dùng đang **đăng nhập**  - Đã lưu token hoặc session |
| **Điều kiện sau** | - Token bị xóa  - Người dùng được chuyển về trang không yêu cầu đăng nhập |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện đăng xuất) * **Backend:** Nodejs (xử lý logic) * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Thấp |

### ***2.1.4 PB04-Quên mật khẩu***

|  |  |
| --- | --- |
| **ID** | **PB04** |
| **Tiêu đề** | Quên mật khẩu |
| **Mô tả** | Người dùng khôi phục mật khẩu khi quên bằng cách xác thực qua email và OTP |
| **Các bước thực hiện** | - Người dùng chọn **“Quên mật khẩu?”** tại giao diện đăng nhập  - Nhập **email đã đăng ký** → nhấn **Gửi yêu cầu**  - Hệ thống kiểm tra email:  + Sai định dạng → báo lỗi **"Email không hợp lệ"**  + Đúng → gửi **OTP về email**  - Người dùng nhập **OTP**  + Sai OTP → báo lỗi  + Đúng → cho phép đổi mật khẩu  - Nhập **mật khẩu mới + xác nhận mật khẩu**  + Nếu không trùng → báo lỗi  + Nếu hợp lệ → hệ thống cập nhật mật khẩu mới  - Hiển thị thông báo: **“Đổi mật khẩu thành công”** và chuyển về **đăng nhập** |
| **Điều kiện trước** | - Người dùng đã có **tài khoản tồn tại**  - Đang **truy cập website** |
| **Điều kiện sau** | - Mật khẩu mới được cập nhật thành công  - Người dùng có thể đăng nhập lại bằng mật khẩu mới |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện Quên mật khẩu) * **Backend:** Nodejs (xử lý logic) * **Database**: Mongodb * **Email Service** : Email API * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình - Cao |

### ***2.1.5 PB05-Tìm kiếm khóa học***

|  |  |
| --- | --- |
| **ID** | **PB05** |
| **Tiêu đề** | Tim kiếm khóa học |
| **Mô tả** | Người dùng tìm kiếm khóa học trên website theo tên hoặc mã khóa học |
| **Các bước thực hiện** | - Người dùng truy cập **trang chủ**  - Có thể tìm kiếm theo 2 hình thức:  **+ Tìm theo tiêu chí có sẵn** (Tên hoặc mã khóa học)  \* Chọn tiêu chí từ **ô tìm kiếm**  \* Nhập thông tin vào ô tương ứng  \* Nhấn tìm kiếm  - Hệ thống xử lý:  + Nếu thông tin **tồn tại trong CSDL** → hiển thị kết quả tìm kiếm  + Nếu không có kết quả → hiển thị thông báo: **“Không tìm thấy”** |
| **Điều kiện trước** | - Người dùng đã truy cập **trang chủ**  - Hệ thống có **dữ liệu khóa học** trong cơ sở dữ liệu |
| **Điều kiện sau** | - Kết quả tìm kiếm được hiển thị nếu có  - Hoặc hiển thị thông báo nếu không tìm thấy dữ liệu |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện tìm kiếm , filter) * **Backend:** Nodejs (xử lý logic ) * **Database**: Mongodb * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình - Thấp |

### ***2.1.6 PB06-Quản lý thông tin cá nhân***

|  |  |
| --- | --- |
| **ID** | **PB06** |
| **Tiêu đề** | Quản lý thông tin cá nhân |
| **Mô tả** | Người dùng xem và chỉnh sửa thông tin cá nhân của mình trên hệ thống |
| **Các bước thực hiện** | - Người dùng truy cập **trang website**  - Chọn mục **“Thông tin cá nhân”** hoặc **“Hồ sơ cá nhân”**  - Hệ thống hiển thị các thông tin cá nhân:  + Tên, email, số điện thoại, địa chỉ, ngày sinh, giới tính,...  - Người dùng có thể **chỉnh sửa thông tin**  - Nhấn nút **“Lưu”** để cập nhật  - Hệ thống xác thực và cập nhật thông tin:  + Nếu hợp lệ → thông báo **“Cập nhật thành công”**  + Nếu có lỗi → hiển thị thông báo tương ứng (ví dụ: "Email không hợp lệ") |
| **Điều kiện trước** | - Người dùng đã **đăng nhập**  - Có **tài khoản tồn tại** trong hệ thống |
| **Điều kiện sau** | - Thông tin cá nhân được hiển thị đúng  - Người dùng có thể cập nhật thông tin thành công |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện hồ sơ người dùng) * **Backend:** Nodejs (xử lý logic ) * **Database**: Mongodb * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình |

### ***2.1.7 PB07-Quản lý khóa học***

|  |  |
| --- | --- |
| **ID** | **PB07** |
| **Tiêu đề** | Quản lý thông tin cá nhân |
| **Mô tả** | Admin quản lý danh sách khóa học trên hệ thống: xem, thêm, sửa, xóa |
| **Các bước thực hiện** | - Admin **đăng nhập** vào hệ thống  + Chọn chức năng **“Quản lý khóa học”**  - Hệ thống hiển thị giao diện quản lý khóa học với các tùy chọn: \* Xem khóa học: Hiển thị ****danh sách tất cả khóa học**** đang có\* Thêm khóa học: + Hiển thị **form thêm mới**  + Admin nhập thông tin và nhấn **Lưu**  + Hệ thống thêm khóa học mới vào danh sách \* Sửa khóa học: + Hiển thị form chỉnh sửa  + Admin cập nhật thông tin → nhấn **Lưu**  + Hệ thống cập nhật dữ liệu khóa học \* Xóa khóa học: + Hiển thị xác nhận xóa  + Admin xác nhận → hệ thống **xóa khóa học** |
| **Điều kiện trước** | - Admin đã đăng nhập thành công |
| **Điều kiện sau** | - Danh sách khóa học được hiển thị  - Admin có thể **xem, thêm, sửa, xóa** khóa học thành công |
| **Công nghệ sử dụng** | * **Frontend:** React.js (Bảng danh sách , form thêm / sửa) * **Backend:** Nodejs (xử lý logic ) * **Database**: Mongodb * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình - Cao |

### ***2.1.8 PB08-Xem lịch học***

|  |  |
| --- | --- |
| **ID** | **PB08** |
| **Tiêu đề** | Xem lịch học |
| **Mô tả** | Học viên vào website để xem lịch học của mình |
| **Các bước thực hiện** | - Người dùng truy cập **trang website**  - Chọn mục **“Lịch học”**  - Hệ thống hiển thị lên lịch học của người dùng |
| **Điều kiện trước** | - Người dùng đã **đăng nhập**  - Có **tài khoản tồn tại** trong hệ thống |
| **Điều kiện sau** | - Lịch học được hiển thị đúng |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện lịch học) * **Backend:** Nodejs (xử lý logic ) * **Database**: Mongodb * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình |

### ***2.1.9 PB09-Xem lịch dạy***

|  |  |
| --- | --- |
| **ID** | **PB09** |
| **Tiêu đề** | Xem lịch dạy |
| **Mô tả** | Giảng viên vào website để xem lịch dạy của mình |
| **Các bước thực hiện** | - Người dùng truy cập **trang website**  - Chọn mục **“Lịch dạy”**  - Hệ thống hiển thị lên lịch dạy của người dùng |
| **Điều kiện trước** | - Người dùng đã **đăng nhập**  - Có **tài khoản tồn tại** trong hệ thống |
| **Điều kiện sau** | - Lịch dạy được hiển thị đúng |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện lịch học) * **Backend:** Nodejs (xử lý logic ) * **Database**: Mongodb * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình |

### ***2.1.10 PB10-Quản lý giỏ hàng***

|  |  |
| --- | --- |
| **ID** | **PB10** |
| **Tiêu đề** | Quản lý giỏ hàng |
| **Mô tả** | Học viên vào website để xem và thao tác trong giỏ hàng |
| **Các bước thực hiện** | - Học viên đăng nhập thành công vào website  - Học viên bấm chọn vào mục giỏ hàng  - Hệ thống chuyển đến trang giỏ hàng và hiển thị danh sách các khóa học có trong giỏ hàng  - Học viên muốn thêm khóa học vào giỏ hàng thì ở trang chi tiết khóa học chọn thêm vào giỏ hàng. Sau đó chọn lớp học và ấn thêm vào giỏ hàng . Hệ thống sẽ cập nhật lại giỏ hàng  - Học viên muốn xóa khóa học thì thực hiện chọn xóa khóa học. Hệ thống sẽ cập nhật lại giỏ hàng  - Học viên muốn thanh toán khóa học thì nhấn nút thanh toán. |
| **Điều kiện trước** | - Người dùng đã **đăng nhập**  - Có **tài khoản tồn tại** trong hệ thống |
| **Điều kiện sau** | - Giỏ hàng được cập nhật chính xác theo các thao tác. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện lịch học) * **Backend:** Nodejs (xử lý logic ) * **Database**: Mongodb * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình - Cao |

### ***2.1.11 PB11-Quản lý thanh toán***

|  |  |
| --- | --- |
| **ID** | **PB11** |
| **Tiêu đề** | Quản lý thanh toán |
| **Mô tả** | Học viên vào website truy cập trang giỏ hàng và thực hiện thanh toán khóa học |
| **Các bước thực hiện** | - Admin đăng nhập thành công vào hệ thống.  - Admin chọn mục “**Quản lý thanh toán**”  - Hệ thống hiển thị danh sách các hóa đơn/thanh toán.  - Admin chọn một hóa đơn để xem chi tiết.  - Hệ thống hiển thị đầy đủ thông tin hóa đơn. |
| **Điều kiện trước** | - Admin đã đăng nhập thành công |
| **Điều kiện sau** | - Admin xem được chính xác thông tin thanh toán. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện lịch học) * **Backend:** Nodejs (xử lý logic ) * **Database**: Mongodb * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình - Cao |

### ***2.1.12 PB12-Quản lý tài khoản***

|  |  |
| --- | --- |
| **ID** | **PB12** |
| **Tiêu đề** | Quản lý tài khoản |
| **Mô tả** | Admin vào website để quản lý tài khoản của học viên và giảng viên |
| **Các bước thực hiện** | - Admin đăng nhập thành công vào hệ thống  - Chọn chức năng *quản lý tài khoản*  - Hệ thống hiển thị giao diện quản lý tài khoản  \* Admin chọn xem   * Hệ thống hiện thị danh sách tài khoản   \* Admin chọn thêm   * Hệ thống hiển thị giao diện thêm tài khoản. * Admin cập nhật thông tin vào form, ấn Cập nhật.   \* Admin chọn xóa   * Hệ thống hiển thị danh sách tài khoản. * Admin chọn tài khoản muốn xoá, ấn Xoá.   - Hệ thống kiểm tra :  \* Nếu thành công thông báo “ Cập nhật thành công”.  - Nếu không thành công thông báo “ Cập nhật không thành công”. |
| **Điều kiện trước** | - Admin đã đăng nhập thành công |
| **Điều kiện sau** | - Danh sách tài khoản người dùng được cập nhật chính xác. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện lịch học) * **Backend:** Nodejs (xử lý logic ) * **Database**: Mongodb * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình - Cao |

### ***2.1.13 PB13-Quản lý lớp học***

|  |  |
| --- | --- |
| **ID** | **PB13** |
| **Tiêu đề** | Quản lý lớp học |
| **Mô tả** | Admin vào website quản lý các lớp học để đảm bảo dữ liệu lớp học luôn được cập nhật đầy đủ và chính xác. |
| **Các bước thực hiện** | - Admin đăng nhập thành công vào hệ thống.  - Chọn mục **Quản lý lớp học** trong dashboard.  - Hệ thống hiển thị danh sách các lớp học hiện tại.  - Admin có thể thực hiện:  + Thêm lớp → Nhập thông tin → Lưu → Thông báo thành công → Danh sách cập nhật.  + Sửa lớp → Chọn lớp → Sửa thông tin → Lưu → Thông báo thành công → Danh sách cập nhật.  + Xóa lớp → Chọn lớp → Xác nhận xóa → Hệ thống cập nhật danh sách.  - Tất cả thay đổi được phản ánh tức thì trong giao diện. |
| **Điều kiện trước** | - Admin đã đăng nhập thành công |
| **Điều kiện sau** | - Danh sách lớp học được hiển thị và cập nhật đúng theo thao tác. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện lịch học) * **Backend:** Nodejs (xử lý logic ) * **Database**: Mongodb * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình - Cao |

### ***2.1.14 PB14-Quản lý bài thi***

|  |  |
| --- | --- |
| **ID** | **PB14** |
| **Tiêu đề** | Quản lý bài thi |
| **Mô tả** | Admin vào website quản lý danh sách bài thi trong hệ thống: thêm mới, sửa đổi, xóa hoặc theo dõi thông tin bài thi của từng lớp/học viên. |
| **Các bước thực hiện** | - Admin đăng nhập vào hệ thống.  - Truy cập mục **Quản lý bài thi**.  - Hệ thống hiển thị danh sách bài thi hiện có.  - Admin có thể chọn thao tác  \* Thêm bài thi → Nhập thông tin → Lưu → Thông báo và cập nhật danh sách.  \* Sửa bài thi → Chọn bài → Chỉnh sửa → Lưu → Thông báo và cập nhật danh sách.  \* Xóa bài thi → Chọn bài → Xác nhận xóa → Cập nhật danh sách. |
| **Điều kiện trước** | - Admin đã đăng nhập thành công |
| **Điều kiện sau** | - Danh sách bài thi được cập nhật và hiển thị chính xác.  - Các thao tác thêm/sửa/xóa không làm ảnh hưởng dữ liệu khác. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện lịch học) * **Backend:** Nodejs (xử lý logic ) * **Database**: Mongodb * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình - Cao |

* 1. Phân tích về Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint** | **Function** | **Started Date** | **Due Date** |
| Sprint 1 | Develop | 10/03/2025 | 29/03/2025 |

1. HẠN CHẾ

|  |  |
| --- | --- |
| **Hạn chế** | **Tình trạng** |
| **Thời gian** | Thời gian để hoàn thành dự án là 614 giờ. |
| **Nhân lực** | Toàn bộ nhóm, tổng cộng 5 người làm việc cùng nhau để hoàn thành dự án. |
| **Các yêu cầu** | Dựa trên các yêu cầu của Product owner. |

1. TÓM LƯỢC

|  |  |  |
| --- | --- | --- |
| **Name** | **Mô tả** | **Role** |
| **Product Owner** | Người đưa ra yêu cầu. | Cung cấp thông tin để phát triển hệ thống. Ra quyết định nghiệm thu và triển khai dự án, làm unit test. |
| **Scrum Master** | Đây là bên liên quan lãnh đạo, quản lý Nhóm phát triển hệ thống. | Kiểm soát, quản lý, giám sát, đảm bảo dự án hoàn thành đúng thời hạn, ngân sách, đúng kế hoạch và yêu cầu. |
| **Requirement Analyzer** | Đây là một bên liên quan làm việc với các Nhà phân tích để truyền đạt chính xác các yêu cầu hoặc nhu cầu thành các yêu cầu được sử dụng cho thiết kế. | Chỉ định chi tiết của một hoặc nhiều phần chức năng của hệ thống bằng cách mô tả một hoặc các khía cạnh của yêu cầu.  Điều này sẽ bao gồm chức năng và phi chức năng. |
| **Software Architect** | Đây là một bên liên quan nắm vai trò chính, dẫn dắt sự phát triển hệ thống. | Chịu trách nhiệm về kiến trúc phần mềm, bao gồm các quyết định kỹ thuật quan trọng hạn chế thiết kế tổng thể cho dự án. |
| **Coder** | Đây là bộ phận lập trình phần mềm. | Thực hiện dự án. |