**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 Gia Khang | | | | | | | |
| **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 Gia Khang | 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 |

MỤC LỤC

[1. GIỚI THIỆU 1](#_Toc3231)

[1.1. Mục đích 1](#_Toc27318)

[1.2. Phạm vi 1](#_Toc1823)

[1.3. Tham khảo 2](#_Toc20553)

[1.4. Biểu đồ UseCase Tổng quát 3](#_Toc4681)

[2. PRODUCT BACKLOG 4](#_Toc18881)

[2.1. Danh sách các Product Backlog trong Hệ thống 4](#_Toc26908)

*[2.1.1 PB01-Đăng ký 8](#_Toc16180)*

*[2.1.2 PB02-Đăng nhập 9](#_Toc20515)*

*[2.1.3 PB03-Tìm kiếm khóa học](#_Toc5279)* [12](#_Toc5279)

*[2.1.4 PB04-Xem chi tiết khóa học](#_Toc3711)* [13](#_Toc3711)

*[2.1.5 PB05-Mua khóa học](#_Toc852)* [14](#_Toc852)

*[2.1.6 PB06-Thanh toán](#_Toc5156)* [15](#_Toc5156)

*[2.1.7 PB07-Xem lịch sử mua hàng](#_Toc4025)* [16](#_Toc4025)

*[2.1.8 PB08-Xem lịch học](#_Toc24348)* [17](#_Toc24348)

*[2.1.9 PB09-Nhận thông báo](#_Toc1539)* [17](#_Toc1539)

*[2.1.10 PB10-Xem bài viết](#_Toc32718)* [18](#_Toc32718)

*[2.1.11 PB11-Làm và xem kết quả thi](#_Toc32218)* [19](#_Toc32218)

*[2.1.12 PB12-Đánh giá khóa học](#_Toc23711)* [20](#_Toc23711)

*[2.1.13 PB13-Quản lý thông tin cá nhân](#_Toc9222)* [21](#_Toc9222)

*[2.1.14 PB14-Điểm danh](#_Toc5984)* [22](#_Toc5984)

*[2.1.15 PB15-Xem lịch dạy](#_Toc17043)* [23](#_Toc17043)

*[2.1.16 PB16-Quản lý bài thi](#_Toc19303)* [23](#_Toc19303)

*[2.1.17 PB17-Quản lý điểm](#_Toc786)* [24](#_Toc786)

*[2.1.18 PB18-Quản lý khóa học](#_Toc20812)* [25](#_Toc20812)

*[2.1.19 PB19-Quản lý lớp](#_Toc990)* [27](#_Toc990)

*[2.1.20 PB20-Quản lý học viên](#_Toc23960)* [28](#_Toc23960)

*[2.1.21 PB21-Quản lý giảng viên](#_Toc24718)* [29](#_Toc24718)

*[2.1.22 PB22-Quản lý thanh toán](#_Toc14827)* [31](#_Toc14827)

*[2.1.23 PB23-Quản lý bài viết](#_Toc3841)* [31](#_Toc3841)

*[2.1.24 PB24-Quản lý đánh giá](#_Toc5227)* [33](#_Toc5227)

*[2.1.25 PB25-Báo cáo thống kê](#_Toc15771)* [34](#_Toc15771)

[2.2. Phân tích về Sprint Backlog 35](#_Toc32624)

[3. HẠN CHẾ 35](#_Toc1205)

[4. TÓM LƯỢC 35](#_Toc19401)

# 

# 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

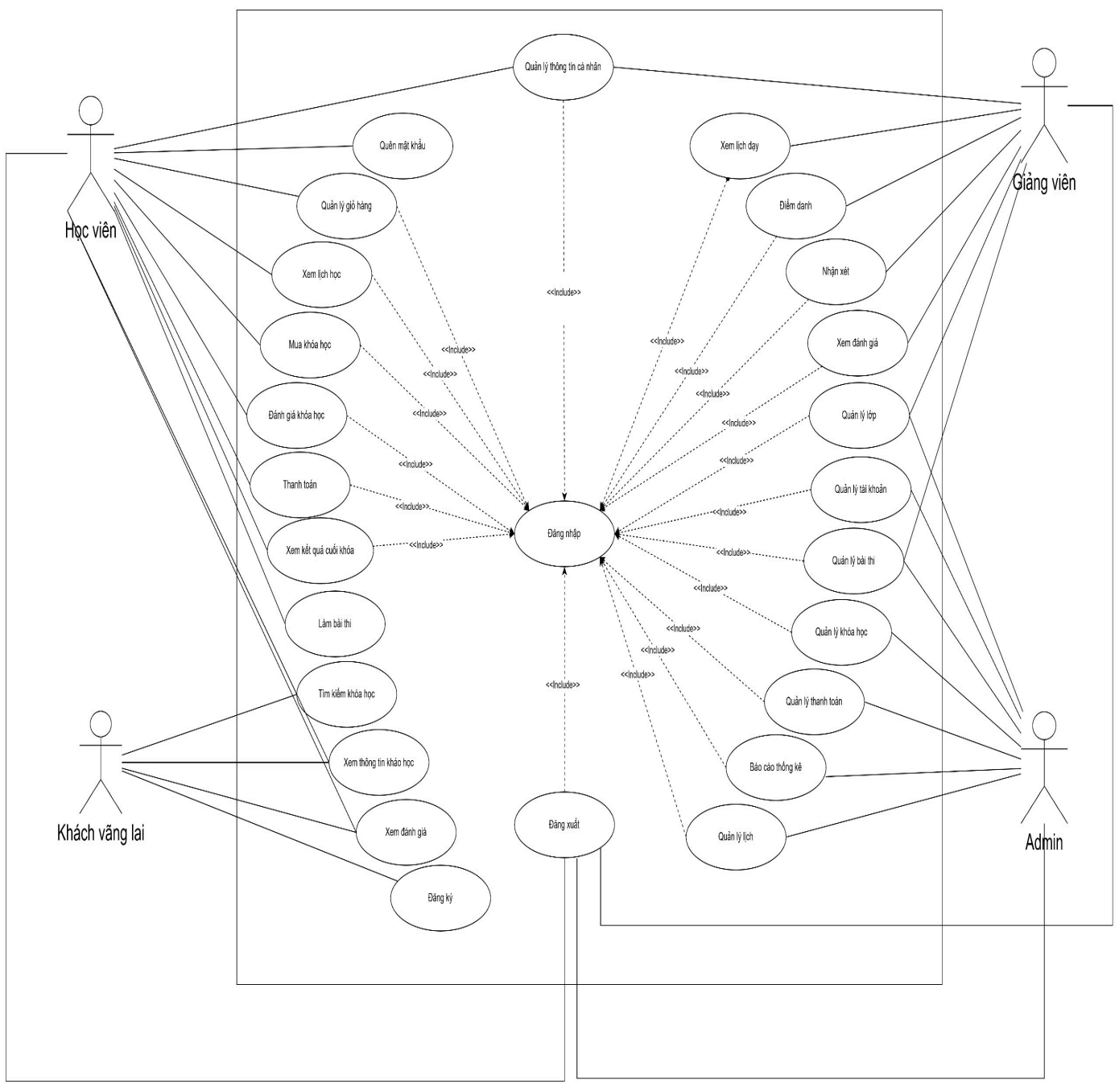
Đề tài tập trung vào việc xây dựng một hệ thống phần mềm quản lý trung tâm dạy học cờ vua tích hợp AI tư vấn khách hàng, nhằm số hóa quy trình quản lý vận hành và nâng cao trải nghiệm người dùng thông qua công nghệ. Phạm vi thực hiện Hệ thống sẽ bao gồm các chức năng chính sau: Quản lý khóa học: Thêm, 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 ký | Khách vãng lai | Tạo tài khoản | Khách vãng lai có thể đăng ký tài khoản | 1 |
| PB02 | Đăng nhập | Học viên/Giảng viên/Admin | Truy cập vào hệ thống học tập | Người dùng có thể đăng nhập vào hệ thống . Có thể lấy lại mật khẩu nếu quên và đăng xuất khi đã đăng nhập vào hệ thống | 1 |
| PB03 | Tìm kiếm khóa học | Khách vãng lai/Học viên | Tìm khóa học phù hợp | Người dùng có thể tìm khóa học theo tên, trình độ khóa học | 1 |
| PB04 | Xem chi tiết khóa học | Khách vãng lai /Học viên/Giảng viên/Admin | Xem chi tiết khóa học | Người dùng có thể xem chi tiết các khóa học trong hệ thống | 2 |
| PB05 | Mua khóa học | Học viên | Mua khóa học | Học viên có thể mua khóa học đã thêm vào giỏ hàng | 2 |
| PB06 | Thanh toán | Học viên | Hoàn tất giao dịch mua khóa học | Người dùng có thể thanh toán khóa học một cách an toàn | 1 |
| PB07 | Xem lịch sử mua hàng | Học viên | Xem lại các khóa học đã mua (đăng ký) | Học viên có thể mở để xem các khóa học đã mua (đăng ký) trước đó. | 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 | Nhận thông báo | Học viên | Nhận được các thông báo | Học viên có thể nhận được các thông báo như thanh toán , điểm danh … Qua email | 3 |
| PB10 | Xem bài viết | Học viên / Khách vãng lai / Giảng viên / Admin | Xem các bài viết của trung tâm | Người dùng có thể xem các bài viết nổi bật về trung tâm | 2 |
| PB11 | Làm và xem kết quả thi | Học viên | Làm bài thi , kiểm tra và xem kết quả | Học viên có thể làm bài thi cuối khóa và xem kết quả thi. | 1 |
| PB12 | 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 |
| PB13 | Quản lý thông tin cá nhân | Học viên / Giảng viên / Admin | Quản lý các thông tin cá nhân | Người dùng có thể quản lý các thông tin như họ tên , SĐTT , địa chỉ …. | 1 |
| PB14 | Điểm danh | Giảng viên | Ghi nhận tình trạng tham gia của học viên | Giáo viên có thể điểm danh theo buổi học và sẽ thông báo về email | 1 |
| PB15 | Xem lịch dạy | Giảng viên | Xem lịch giảng dạy | Giáo viên có thể xem lịch các lớp mình giảng dạy | 1 |
| PB16 | Quản lý bài thi | Giảng viên / Admin | Quản lý đề thi, điểm và lịch thi | Có thể thêm, sửa, phân phối bài thi | 1 |
| PB17 | Quản lý điểm | Giảng viên | Quản lý điểm thi của học viên | Giảng viên quản lý điểm thi của học viên theo từng lớp | 1 |
| PB18 | Quản lý khóa học | Admin | Quản lý nội dung khóa học | Có thể thêm, sửa, xóa khóa học | 1 |
| PB19 | Quản lý lớp | Admin | Quản lý danh sách lớp học | Có thể thêm, chỉnh sửa, xóa lớp học | 1 |
| PB20 | Quản lý học viên | Admin | Quản lý danh sách học viên | Có thể xem , xóa tài khoản học viên đó | 1 |
| PB21 | Quản lý giảng viên | Admin | Quản lý danh sách các giảng viên | Có thể tạo tài khoản cho giảng viên và xóa đi tài khoản không cần thiết | 1 |
| PB22 | Quản lý thanh toán | Admin | Quản lý các hóa đơn đăng ký khóa học đã thanh toán | Quản lý các danh sách các hóa đơn khóa học đã thanh toán . | 1 |
| PB23 | Quản lý bài viết | Admin | Quản lý các bài viết của trung tâm | Quản lý danh sách các bài viết của trung tâm . Có thể thêm, sửa , xóa bài viết. | 1 |
| PB24 | Quản lý đánh giá | Admin | Theo dõi và kiểm duyệt đánh giá khóa học | Có thể xem và xóa các đánh giá của giảng viên | 2 |
| PB25 | Báo cáo 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 |

***\*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 |

#### *2.1.2.1 PB02.1-Đăng xuất*

|  |  |
| --- | --- |
| **ID** | **PB02.1** |
| **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 |

#### *2.1.2.2 PB02.2-Quên mật khẩu*

|  |  |
| --- | --- |
| **ID** | **PB02.2** |
| **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 |

### ***2.1.3 PB03-Tìm kiếm khóa học***

|  |  |
| --- | --- |
| **ID** | **PB03** |
| **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 |

### ***2.1.4 PB04-Xem chi tiết khóa học***

|  |  |
| --- | --- |
| **ID** | **PB04** |
| **Tiêu đề** | Xem chi tiết khóa học |
| **Mô tả** | Người dùng có thể xem thông tin chi tiết của khóa học bao gồm: giá thành, thời gian bắt đầu, thời gian kết thúc, giảng viên, số lượng chỗ còn trống. |
| **Các bước thực hiện** | - Người dùng truy cập vào trang danh sách khóa học  - Chọn một khóa học muốn xem chi tiết  - Hệ thống xử lý yêu cầu và hiển thị thông tin chi tiết khóa học bao gồm:  + Tên khóa học  + Giá khóa học  + Thời gian bắt đầu – kết thúc  + Tên giảng viên phụ trách  + Số lượng chỗ còn trống  - Người dùng xem thông tin  - Kết thúc quá trình xem |
| **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** | - Thông tin chi tiết của khóa học được hiển thị đầy đủ |
| **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 |

### ***2.1.5 PB05-Mua khóa học***

|  |  |
| --- | --- |
| **ID** | **PB05** |
| **Tiêu đề** | Mua khóa học |
| **Mô tả** | Học viên sau khi đăng nhập vào hệ thống có thể thực hiện thao tác mua khóa học. |
| **Các bước thực hiện** | - Học viên đăng nhập vào hệ thống  - Truy cập danh sách khóa học  - Chọn một khóa học muốn mua  - Hệ thống hiển thị chi tiết thông tin khóa học  - Học viên nhấn nút “Mua khóa học”  - Hệ thống chuyển sang giao diện chọn lớp học  - Học viên chọn một lớp học phù hợp với lịch trình  - Hệ thống cập nhật giỏ hàng với khóa học đã chọn  - Kết thúc quá trình mua |
| **Điều kiện trước** | - Người dùng đã **đăng nhập**  - Có **tài khoản tồn tại** trong hệ thống |
| **Điều kiện sau** | - Khóa học được thêm vào giỏ hàng của học viên 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 |

### ***2.1.6 PB06-Thanh toán***

|  |  |
| --- | --- |
| **ID** | **PB06** |
| **Tiêu đề** | Thanh toán |
| **Mô tả** | Học viên thực hiện thanh toán cho các khóa học đã chọn trong giỏ hàng bằng phương thức VNPay. |
| **Các bước thực hiện** | - Học viên đăng nhập vào hệ thống  - Truy cập giao diện giỏ hàng  - Chọn khóa học cần thanh toán và nhấn nút “**Thanh toán**”  - Hệ thống hiển thị giao diện thanh toán  - Học viên chọn phương thức thanh toán là VNPay  - Hệ thống hiển thị giao diện tích hợp VNPay  - Học viên nhập thông tin ngân hàng  - Hệ thống thực hiện kiểm tra:  + Thông tin ngân hàng hợp lệ  + Khóa học chưa được đăng ký trước đó  - Nếu hợp lệ:  + Gửi email xác nhận thanh toán thành công  + Hiển thị thông báo thành công  - Nếu không hợp lệ:  + Hiển thị thông báo thanh toán thất bại  - Kết thúc quá trình |
| **Điều kiện trước** | - Học viên đã đăng nhập vào hệ thống  - Trong giỏ hàng có khóa học hợp lệ để thanh toán |
| **Điều kiện sau** | - Thanh toán được xác nhận thành công hoặc thất bại và thông báo cho người dùng |
| **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 |

***2.1.7 PB07-Xem lịch sử mua hàng***

|  |  |
| --- | --- |
| **ID** | **PB07** |
| **Tiêu đề** | Xem lịch sử mua hàng |
| **Mô tả** | Học viên có thể truy cập vào mục “Khóa học của tôi” để xem danh sách các khóa học đã mua/đăng ký trước đó, |
| **Các bước thực hiện** | - Học viên đăng nhập vào hệ thống  - Chọn mục “Khóa học của tôi” trong giao diện người dùng  - Hệ thống hiển thị danh sách các khóa học mà học viên đã đăng ký hoặc thanh toán thành công  - Học viên xem chi tiết danh sách khóa học đã mua  - Kết thúc quá trình |
| **Điều kiện trước** | - Học viên đã đăng nhập vào hệ thống  - Hệ thống đã ghi nhận các khóa học học viên đã mua |
| **Điều kiện sau** | - Danh sách các khóa học đã đăng ký được hiển thị thành công. |
| **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 |

### ***2.1.8 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-Nhận thông báo***

|  |  |
| --- | --- |
| **ID** | **PB09** |
| **Tiêu đề** | Nhận thông báo |
| **Mô tả** | Học viên sẽ nhận được các thông báo quan trọng từ quản trị viên (admin) qua email. |
| **Các bước thực hiện** | - Hệ thống tự động gửi email đến địa chỉ email của học viên  - Học viên kiểm tra hộp thư để đọc nội dung thông báo  - Kết thúc quá trình |
| **Điều kiện trước** | - Hệ thống đã có địa chỉ email hợp lệ của học viên |
| **Điều kiện sau** | - Học viên nhận được email thông báo thành công từ hệ thống |
| **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-Xem bài viết***

|  |  |
| --- | --- |
| **ID** | **PB10** |
| **Tiêu đề** | Xem bài viết |
| **Mô tả** | Học viên có thể truy cập vào website để xem các bài viết, tin tức, thông báo hoặc tài liệu học tập được đăng bởi trung tâm (admin). |
| **Các bước thực hiện** | - Học viên truy cập vào trang web của trung tâm  - Hệ thống hiển thị danh sách các bài viết do admin đăng  - Học viên chọn và đọc bài viết tùy ý  - Kết thúc quá trình. |
| **Điều kiện trước** | - Không có (không yêu cầu đăng nhập) |
| **Điều kiện sau** | - Danh sách bài viết hiển thị thành công, học viên có thể xem nội dung |
| **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** | Thấp |

### ***2.1.11 PB11-Làm và xem kết quả thi***

|  |  |
| --- | --- |
| **ID** | **PB11** |
| **Tiêu đề** | Làm và xem kết quả thi |
| **Mô tả** | Học viên có thể chọn và làm bài thi trên hệ thống, sau đó hệ thống sẽ lưu kết quả. |
| **Các bước thực hiện** | - Làm bài thi:  + Học viên đăng nhập, chọn bài thi muốn thực hiện  + Hệ thống hiển thị danh sách bài thi có sẵn  + Học viên chọn nút “Làm bài”  + Hệ thống hiển thị đề thi  + Học viên làm bài và chọn “Submit” để nộp  + Hệ thống cập nhật kết quả vào cơ sở dữ liệu  + Kết thúc quá trình.  - Xem kết quả bài thi:  + Học viên chọn bài thi muốn xem kết quả  + Hệ thống hiển thị kết quả của bài thi đó  + Học viên xem chi tiết kết quả  + Kết thúc quá trình. |
| **Điều kiện trước** | - Học viên đã đăng nhập vào hệ thống  - Hệ thống đã có danh sách bài thi |
| **Điều kiện sau** | - Kết quả được lưu lại sau khi làm bài  - Học viên có thể xem lại kết quả bất kỳ lúc nào |
| **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-Đánh giá khóa học***

|  |  |
| --- | --- |
| **ID** | **PB12** |
| **Tiêu đề** | Đánh giá khóa học |
| **Mô tả** | Học viên có thể đánh giá khóa học sau khi đã hoàn thành. |
| **Các bước thực hiện** | - Học viên đăng nhập, chọn khóa học muốn đánh giá  - Hệ thống kiểm tra trạng thái khóa học:  + Nếu đã hoàn thành → hiển thị giao diện đánh giá  + Nếu chưa hoàn thành → hiển thị thông báo “Chưa đủ điều kiện đánh giá”  - Học viên nhập nội dung đánh giá và chọn nút “Gửi”  - Hệ thống cập nhật đánh giá vào cơ sở dữ liệu  - Kết thúc quá trình. |
| **Điều kiện trước** | - Học viên đã đăng nhập  - Học viên đã hoàn thành khóa học |
| **Điều kiện sau** | - Đánh giá được lưu trữ trong hệ thống  - Có thể hiển thị đánh giá đó cho các người dùng 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** | Thấp - Trung bình |

### ***2.1.13 PB13-Quản lý thông tin cá nhân***

|  |  |
| --- | --- |
| **ID** | **PB13** |
| **Tiêu đề** | Quản lý thông tin cá nhân |
| **Mô tả** | Chức năng cho phép Học viên và Giảng viên xem, chỉnh sửa và cập nhật các thông tin cá nhân như tên, số điện thoại, email, địa chỉ, ảnh đại diện,... trong hệ thống. |
| **Các bước thực hiện** | - Luồng 1: Xem thông tin cá nhân  + Người dùng chọn mục "Quản lý thông tin cá nhân"  + Hệ thống hiển thị giao diện với đầy đủ thông tin cá nhân  + Người dùng xem thông tin cá nhân  + Kết thúc quá trình.  - Luồng 2: Cập nhật thông tin cá nhân  + Người dùng chỉnh sửa thông tin cá nhân và nhấn “Cập nhật”  + Hệ thống kiểm tra, cập nhật thông tin mới vào cơ sở dữ liệu  + Hiển thị thông báo thành công/thất bại  + Kết thúc quá trình. |
| **Điều kiện trước** | - Người dùng đã đăng nhập vào hệ thống thành công |
| **Điều kiện sau** | - Hệ thống lưu lại thông tin cá nhân đã được chỉnh sửa  - Thông tin hiển thị là mới nhất |
| **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 |

### ***2.1.14 PB14-Điểm danh***

|  |  |
| --- | --- |
| **ID** | **PB14** |
| **Tiêu đề** | Điểm danh |
| **Mô tả** | Giảng viên thực hiện điểm danh học viên trong lớp. Sau khi điểm danh, hệ thống tự động gửi email thông báo tình hình đi học (có mặt/vắng mặt) của học viên đến email phụ huynh đã đăng ký. |
| **Các bước thực hiện** | - Giảng viên chọn chức năng “**Điểm danh**”  - Hệ thống hiển thị danh sách các lớp mà giảng viên phụ trách  - Giảng viên chọn lớp cần điểm danh  - Hệ thống hiển thị danh sách học viên trong lớp đó  - Giảng viên đánh dấu tình trạng đi học (có mặt/vắng mặt) cho từng học viên  - Hệ thống:  + Gửi email thông báo cho phụ huynh học viên về tình trạng đi học  + Cập nhật thông tin điểm danh vào cơ sở dữ liệu  - Kết thúc quá trình |
| **Điều kiện trước** | - Giảng viên đã đăng nhập vào hệ thống thành công |
| **Điều kiện sau** | - CSDL đã được cập nhật thông tin điểm danh  - Email thông báo đã được gửi đến phụ huynh |
| **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 |

### ***2.1.15 PB15-Xem lịch dạy***

|  |  |
| --- | --- |
| **ID** | **PB15** |
| **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 |

### ***2.1.16 PB16-Quản lý bài thi***

|  |  |
| --- | --- |
| **ID** | **PB16** |
| **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 |

### ***2.1.17 PB17-Quản lý điểm***

|  |  |
| --- | --- |
| **ID** | **PB17** |
| **Tiêu đề** | Quản lý điểm |
| **Mô tả** | Giảng viên nhập điểm và xem điểm của học viên trong các lớp học mà họ phụ trách. |
| **Các bước thực hiện** | - **Luồng chính:** Quản lý điểm  + Giảng viên chọn chức năng “Quản lý điểm”  + Hệ thống hiển thị giao diện quản lý điểm  + Giảng viên chọn lớp và tìm kiếm học viên qua thanh tìm kiếm học viên  + Hệ thống hiển thị điểm của học viên trong lớp đó  + Giảng viên xem điểm của học viên  + Kết thúc quá trình  - Luồng sửa điểm:  + Giảng viên chọn lớp cần sửa điểm  + Hệ thống hiển thị danh sách học viên trong lớp đó  + Giảng viên sửa điểm của học viên và chọn “Lưu”  + Hệ thống cập nhật điểm vào cơ sở dữ liệu  + Kết thúc quá trình |
| **Điều kiện trước** | - Giảng viên đã đăng nhập vào hệ thống thành công |
| **Điều kiện sau** | - CSDL đã được cập nhật với điểm mới của học viên |
| **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 |

### ***2.1.18 PB18-Quản lý khóa học***

|  |  |
| --- | --- |
| **ID** | **PB18** |
| **Tiêu đề** | Quản lý khóa học |
| **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 |

### ***2.1.19 PB19-Quản lý lớp***

|  |  |
| --- | --- |
| **ID** | **PB19** |
| **Tiêu đề** | Quản lý lớp |
| **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 |

### ***2.1.20 PB20-Quản lý học viên***

|  |  |
| --- | --- |
| **ID** | **PB20** |
| **Tiêu đề** | Quản lý học viên |
| **Mô tả** | Admin có thể xem và xóa thông tin học viên trong hệ thống quản lý học viên. |
| **Các bước thực hiện** | - **Luồng chính:** Quản lý học viên  + Admin chọn chức năng “Học viên”  + Hệ thống hiển thị giao diện quản lý học viên  + Admin chọn lớp học và học viên cần xem  + Hệ thống hiển thị thông tin chi tiết của học viên  + Admin xem thông tin của học viên  + Kết thúc quá trình  - **Luồng xóa học viên:**  + Admin chọn lớp học và nhấn vào icon xóa bên cạnh học viên cần xóa  + Hệ thống hiển thị thông báo xác nhận xóa học viên  + Admin chọn "Xóa"  + Hệ thống kiểm tra xem học viên có khóa học đang tham gia hay không:  - Nếu có: Thông báo không thể xóa học viên đang tham gia khóa học  - Nếu không: Thông báo xóa thành công và cập nhật CSDL  + Kết thúc quá trình |
| **Điều kiện trước** | - Admin đã đăng nhập thành công |
| **Điều kiện sau** | - Nếu học viên không tham gia khóa học, thông tin học viên sẽ bị xóa khỏi hệ thống và CSDL được cập nhật.  - Nếu học viên tham gia khóa học, không có thay đổi và học viên không bị xóa. |
| **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 |

### ***2.1.21 PB21-Quản lý giảng viên***

|  |  |
| --- | --- |
| **ID** | **PB21** |
| **Tiêu đề** | Quản lý giảng viên |
| **Mô tả** | Admin có thể xem, thêm và xóa thông tin giảng viên trong hệ thống quản lý giảng viên. |
| **Các bước thực hiện** | - **Luồng chính:** Quản lý giảng viên  + Admin chọn chức năng “Giảng viên”  + Hệ thống hiển thị giao diện quản lý giảng viên  - **Luồng xem thông tin giảng viên:**  + Admin chọn lớp và giảng viên cần xem  + Hệ thống hiển thị thông tin chi tiết của giảng viên  + Admin xem thông tin  + Kết thúc quá trình  - **Luồng thêm giảng viên:**  + Admin chọn “Thêm giảng viên”  + Hệ thống hiển thị giao diện nhập thông tin giảng viên  + Admin nhập thông tin và nhấn “Lưu”  + Hệ thống lưu thông tin vào CSDL  + Kết thúc quá trình  - **Luồng xóa giảng viên:**  + Admin chọn lớp và nhấn icon xóa bên cạnh giảng viên cần xóa  + Hệ thống hiển thị thông báo xác nhận xóa  + Admin chọn “Xóa”  + Hệ thống kiểm tra:  - Nếu giảng viên đang dạy khóa học: Hiển thị thông báo “Không thể xóa”  - Nếu không: Thực hiện xóa, cập nhật CSDL, hiển thị thông báo “Xóa thành công”  + Kết thúc Use Case |
| **Điều kiện trước** | - Admin đã đăng nhập thành công |
| **Điều kiện sau** | - Giảng viên được thêm mới hoặc xóa khỏi hệ thống nếu không vi phạm điều kiện ràng buộc (đang dạy khóa học).  - CSDL được cập nhật theo các thay đổi thực hiện. |
| **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 |

### ***2.1.22 PB22-Quản lý thanh toán***

|  |  |
| --- | --- |
| **ID** | **PB22** |
| **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 |

### ***2.1.23 PB23-Quản lý bài viết***

|  |  |
| --- | --- |
| **ID** | **PB23** |
| **Tiêu đề** | Quản lý bài viết |
| **Mô tả** | Admin có quyền xem, thêm, sửa và xóa các bài viết trong hệ thống nhằm quản lý nội dung thông tin chia sẻ trên nền tảng. |
| **Các bước thực hiện** | - **Luồng chính – Truy cập quản lý bài viết**  + Admin chọn mục “Quản lý bài viết”  + Hệ thống hiển thị giao diện danh sách các bài viết  **- Xem bài viết**  + Admin xem danh sách và nội dung bài viết  + Kết thúc use case  - **Thêm bài viết**  + Admin chọn “Thêm bài viết”  + Hệ thống hiển thị giao diện nhập thông tin bài viết mới  + Admin nhập thông tin, chọn “Lưu”  + Hệ thống cập nhật dữ liệu vào CSDL và hiển thị thông báo thành công  + Kết thúc quá trình  - **Sửa bài viết**  + Admin chọn bài viết muốn sửa, nhấn biểu tượng “Sửa”  + Hệ thống hiển thị giao diện chỉnh sửa bài viết  + Admin sửa nội dung, chọn “Lưu”  + Hệ thống cập nhật thông tin vào CSDL  + Kết thúc quá trình  - **Xóa bài viết**  + Admin chọn bài viết, nhấn biểu tượng “Xóa”  + Hệ thống hiển thị thông báo xác nhận xóa  + Admin xác nhận “Xóa”  + Hệ thống cập nhật CSDL, xóa bài viết, hiển thị thông báo xóa thành công  + Kết thúc quá trình |
| **Điều kiện trước** | - Admin đã đăng nhập thành công |
| **Điều kiện sau** | - Bài viết mới được thêm, sửa đổi, hoặc bị xóa đúng theo hành động từ phía Admin.  - Dữ liệu bài viết trong hệ thống luôn 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 |

### ***2.1.24 PB24-Quản lý đánh giá***

|  |  |
| --- | --- |
| **ID** | **PB24** |
| **Tiêu đề** | Quản lý đánh giá |
| **Mô tả** | Admin xem danh sách các đánh giá từ học viên đối với các khóa học nhằm mục đích theo dõi chất lượng giảng dạy và phản hồi từ người học. |
| **Các bước thực hiện** | - Admin chọn mục “**Quản lý đánh giá**”  - Hệ thống hiển thị giao diện danh sách đánh giá theo từng khóa học  - Admin chọn một khóa học cụ thể  - Hệ thống hiển thị các đánh giá tương ứng của khóa học đã chọn  - Admin xem nội dung đánh giá chi tiết  - Kết thúc quá trình |
| **Điều kiện trước** | - Admin đã đăng nhập vào hệ thống thành công. |
| **Điều kiện sau** | - Hệ thống hiển thị đầy đủ các đánh giá theo khóa học đã chọn để phục vụ công tác quản lý chất lượng đào tạo. |
| **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 |

### ***2.1.25 PB25-Báo cáo thống kê***

|  |  |
| --- | --- |
| **ID** | **PB25** |
| **Tiêu đề** | Báo cáo thống kê |
| **Mô tả** | Admin xem các báo cáo thống kê liên quan đến hoạt động của hệ thống như số lượng học viên, lớp học, khóa học, đánh giá, doanh thu,… theo tháng, quý hoặc năm. |
| **Các bước thực hiện** | - Admin chọn mục “Báo cáo thống kê”  - Hệ thống hiển thị giao diện báo cáo thống kê  - Admin chọn thời gian thống kê (tháng, năm, quý)  - Hệ thống hiển thị báo cáo tương ứng với lựa chọn  - Admin xem nội dung báo cáo  - Kết thúc quá trình |
| **Điều kiện trước** | - Admin đã đăng nhập thành công |
| **Điều kiện sau** | - Hệ thống đã hiển thị đầy đủ thông tin báo cáo theo thời gian được chọn. |
| **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** | 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. |