**ĐẠI HỌC DUY TÂN**

**TRƯỜNG KHMT**

**KHOA CÔNG NGHỆ THÔNG TIN**

**--------------🙖🙐✰🙖🙐-------------**

****

**Tên đề tài:**

**XÂY DỰNG SMART LEARNING HUB -WEBSITE**

**QUẢN LÝ TRUNG TÂM PHẦN MỀM ỨNG DỤNG AI**

**(TÀI LIỆU PRODUCT BACKLOG)**

**GVHD:** Th. S Hồ Lê Viết Nin

**Thành Viên**

Trương Thị Ngọc Ánh 27201239366

Nguyễn Thanh Anh 27211234356

Đoàn Văn Huy 27211230499

Nguyễn Hữu Thành 27211220260

Lê Đức Thảo 27211226738

**Đà Nẵng, Tháng 03 năm 2025**

**THÔNG TIN DỰ ÁN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dự án viết tắt** | **SMART LEARNING HUB** | | | |
| **Tên dự án** | Xây Dựng Smart Learning Hub -Website Quản Lý Trung Tâm Phần Mềm Ứng Dụng AI | | | |
| **Thời gian bắt đầu** | 17/03/2025 | **Thời gian kết thúc** | 17/05/2025 | |
| **Lead Institution** | Khoa Công Nghệ Thông Tin, Trường Đại Học Duy Tân | | | |
| **Giáo viên hướng dẫn** | Hồ Lê Viết Nin  Email: hlvnin88@gmail.com  Phone: 0905455246 | | | |
| **Chủ sở hữu dự án & Chi tiết liên hệ** | Hồ Lê Viết Nin  Email: hlvnin88@gmail.com  Phone: 0905455246 | | | |
| **Đối tác** | Duy Tan University | | | |
| **Scrum Master** | Trương Thị Ngọc Ánh | ngocanhtruong178@gmail.com | | 0961779640 |
| **Thành viên nhóm** | Lê Đức Thảo | ducthao2112@gmail.com | | 0385665243 |
| Nguyễn Hữu Thành | nhtthanh633@gmail.com | | 0327908007 |
| Nguyễn Thanh Anh | nguyenthanhanhtp123@gmail.com | | 0906413507 |
| Đoàn Văn Huy | doanvanhuydhdtu@gmail.com | | 0778251152 |

|  |  |  |  |
| --- | --- | --- | --- |
| **TÊN TÀI LIỆU** | | | |
| **Tên dự án** | Xây Dựng Smart Learning Hub -Website Quản Lý Trung Tâm Phần Mềm Ứng Dụng AI | | |
| **Tiêu đề tài liệu** | Product Backlog | | |
| **Người thực hiện** | Trương Thị Ngọc Ánh | | |
| **Ngày** | 28/03/2025 | **Tên tệp:** | 04\_[KLTN-30]Product Backlog.docx |
| **URL** |  | | |
| **Access** | Khoa CNTT | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **LỊCH SỬ BẢN SỬA** | | | |
| **Version** | **Person** | **Date** | **Description** |
| 1.0 | Trương Thị Ngọc Ánh | 27/03/2025 | Tạo tài liệu |
| 1.1 | Trương Thị Ngọc Ánh | 28/03/2025 | Chỉnh sửa |

|  |  |  |  |
| --- | --- | --- | --- |
| **PHÊ DUYỆT TÀI LIỆU**  Cần có các chữ ký sau để phê duyệt tài liệu này | | | |
| **Người hướng dẫn** | Hồ Lê Viết Nin | **Chữ ký:** |  |
| **Ngày:** | 28/03/2025 |
| **Chủ sở hữu** | Hồ Lê Viết Nin | **Chữ ký:** |  |
| **Ngày:** | 28/03/2025 |
| **Scrum master** | Trương Thị Ngọc Ánh | **Chữ ký:** |  |
| **Ngày:** | 28/03/2025 |
| **Thành viên** | Nguyễn Thanh Anh | **Chữ ký:** |  |
| **Ngày:** | 28/03/2025 |
| Lê Đức Thảo | **Chữ ký:** |  |
| **Ngày:** | 28/03/2025 |
| Đoàn Văn Huy | **Chữ ký:** |  |
| **Ngày:** | 28/03/2025 |
| Nguyễn Hữu Thành | **Chữ ký:** |  |
| **Ngày:** | 28/03/2025 |

**MỤC LỤC**

[1. GIỚI THIỆU 1](#_Toc194095953)

[1.1. Mục đích. 1](#_Toc194095955)

[1.2. Phạm vi. 1](#_Toc194095958)

[1.3. Tài liệu tham khảo. 1](#_Toc194095959)

[1.1. Biểu đồ UseCase tổng quát 2](#_Toc194095960)

[2. PRODUCT BACKLOG 2](#_Toc194095961)

[2.1. Danh sách các Product Backlog trong Hệ thống 2](#_Toc194095962)

[2.2. Mô tả Product Backlog 5](#_Toc194095963)

[2.2.1. PB01- Đăng kí 5](#_Toc194095964)

[2.2.2. PB02 - Đăng nhập 6](#_Toc194095965)

[2.2.3. PB03 - Đăng xuất 7](#_Toc194095966)

[2.2.4. PB04 – Quên mật khẩu 9](#_Toc194095967)

[2.2.5. PB05 Thay đổi mật khẩu 10](#_Toc194095968)

[2.2.6. PB06: Quản lý giáo viên 10](#_Toc194095969)

[2.2.7. PB07 - Quản lý nhân viên 11](#_Toc194095970)

[2.2.8. PB08 - Quản lý học viên 12](#_Toc194095971)

[2.2.9. PB09 - Quản lý chi tiết lớp học 13](#_Toc194095972)

[2.2.10. PB010 - Quản lý lịch thi 14](#_Toc194095973)

[2.2.11. PB011 - Quản lý chức vụ 15](#_Toc194095974)

[2.2.12. PB012- Quản lý khóa học 15](#_Toc194095975)

[2.2.13. PB13 - Quản lý lớp học 16](#_Toc194095976)

[2.2.14. PB14 - Quản lý người dùng 17](#_Toc194095977)

[2.2.15. PB15- Phân quyền người dùng 18](#_Toc194095978)

[2.2.16. PB16- Quản lý phiếu chi 18](#_Toc194095979)

[2.2.17. PB17 - Quản lý phiếu thu 19](#_Toc194095980)

[2.2.18. PB18- Quản lý liên hệ 20](#_Toc194095981)

[2.2.19. PB19 - Quản lý bài viết 20](#_Toc194095982)

[2.2.20. PB120 - Quản lý email 21](#_Toc194095983)

[2.2.21. PB21 Xem thông tin khóa học 22](#_Toc194095984)

[2.2.22. PB22 - Thống kê báo cáo hệ thống 23](#_Toc194095985)

[2.2.23. PB23- Ứng dụng AI trong hỗ trợ học tập 23](#_Toc194095986)

[2.2.24. PB024 -Chatbot AI 24](#_Toc194095987)

[2.2.25. PB25-Quản lý Chatbot AI 25](#_Toc194095988)

[2.3. Chia theo từng Sprirnt 26](#_Toc194095989)

[3. HẠN CHẾ 26](#_Toc194095990)

[4. TÓM LƯỢC 26](#_Toc194095991)

# 1. GIỚI THIỆU

## Product Backlog là tài liệu quan trọng trong quy trình Scrum, bao gồm danh sách tất cả các yêu cầu, tính năng, và công việc cần thực hiện để phát triển sản phẩm. Tài liệu này mô tả các chức năng của người dùng (User Stories), giải thích cách thức hoạt động của từng chức năng, độ ưu tiên của chúng và các công nghệ cần sử dụng để hoàn thành công việc.

## 1.1. Mục đích.

## Mục đích của Product Backlog là cung cấp một danh sách chi tiết và có ưu tiên của tất cả các yêu cầu của người dùng cuối đối với sản phẩm. Mỗi yêu cầu trong Product Backlog sẽ được phân tích và chia thành các nhiệm vụ nhỏ hơn, giúp nhóm phát triển hiểu rõ mục tiêu và phương pháp triển khai.

## Tài liệu này cũng bao gồm các tiêu chí chấp nhận cho mỗi yêu cầu, độ ưu tiên của từng tính năng, và các công nghệ sẽ được sử dụng để hoàn thành các nhiệm vụ đó. Điều này giúp đội ngũ phát triển và Product Owner duy trì sự rõ ràng về những gì cần phải thực hiện và đảm bảo rằng các tính năng quan trọng nhất được phát triển trước.

## 1.2. Phạm vi.

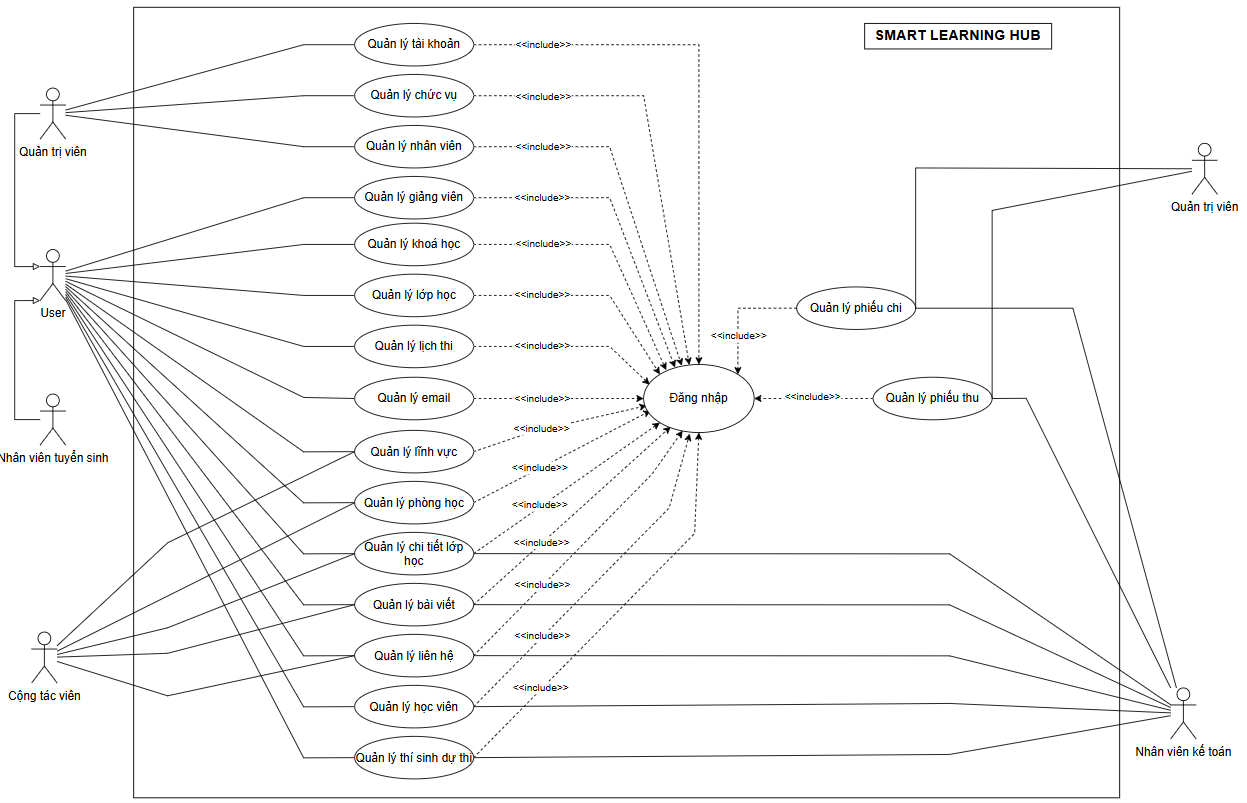
* Các user story trong dự án
* Các yêu cầu chức năng và phi chức năng
* Sơ đồ Use case
* Biểu đồ hoạt động

## 1.3. Tài liệu tham khảo.

Bảng 1: Tài liệu tham khảo

|  |  |  |
| --- | --- | --- |
| **No** | **Reference** | **Note** |
| 1 | <http://agilebench.com/blog/the-product-backlog-for-agile-teams> | Cách tạo Product Backlog |
| 2 | <http://www.mountaingoatsoftware.com/agile/scrum/product-backlog/example/> | Ví dụ về Product Backlog |
| 3 | KLTN-SE.01.Product Proposal.docx | Proposal Document |

* 1. **Biểu đồ UseCase tổng quát**



*Hình 1.1: Biểu đồ Use Case tổng quát*

1. **PRODUCT BACKLOG**
   1. **Danh sách các Product Backlog trong Hệ thống**

*Bảng 2.1: Đặc tả tổng quát các Product Backlog*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Chủ đề** | **Tác nhân** | **Mong muốn** | **Nên** | **Độ ưu tiên** |
| PB01 | Đăng ký | Khách vãng lai | Thêm tài khoản vào hệ thống | Khách hàng có thể đăng ký tài khoản | 3 |
| PB02 | Đăng nhập | Khách hàng / Admin / Nhân viên / Giáo viên | Đăng nhập vào hệ thống | Người dùng có thể đăng nhập vào hệ thống | 2 |
| PB03 | Đăng xuất | Khách hàng / Admin / Nhân viên / Giáo viên | Đăng xuất khỏi hệ thống | Người dùng có thể đăng xuất khỏi phiên truy cập của mình | 3 |
| PB04 | Quên mật khẩu | Khách hàng / Admin / Nhân viên / Giáo viên | Lấy lại mật khẩu để truy cập vào hệ thống | Người dùng có thể lấy lại mật khẩu cho tài khoản của mình | 2 |
| PB05 | Thay đổi mật khẩu | Khách hàng / Admin / Nhân viên / Giáo viên | Đổi mật khẩu hiện tại | Người dùng có thể thay đổi mật khẩu của họ trong hệ thống | 3 |
| PB06 | Quản lý giáo viên | Admin | Quản lý thông tin giáo viên | Admin có thể thêm/sửa/xóa thông tin giáo viên | 1 |
| PB07 | Quản lý nhân viên | Admin | Quản lý thông tin nhân viên | Admin có thể thêm/sửa/xóa thông tin nhân viên | 1 |
| PB08 | Quản lý học viên | Admin / Nhân viên | Quản lý thông tin học viên | Nhân viên có thể thêm/sửa/xóa thông tin học viên | 1 |
| PB09 | Quản lý chi tiết lớp học | Admin / Giáo viên | Quản lý thông tin chi tiết về lớp học | Người dùng có thể cập nhật danh sách học viên, giáo trình, lịch học | 1 |
| PB10 | Quản lý lịch thi | Admin / Giáo viên | Quản lý và sắp xếp lịch thi | Người dùng có thể lên lịch, chỉnh sửa lịch thi cho học viên | 1 |
| PB11 | Quản lý chức vụ | Admin | Quản lý các chức vụ trong hệ thống | Admin có thể thêm/sửa/xóa vai trò của người dùng | 1 |
| PB12 | Quản lý khóa học | Admin / Giáo viên | Quản lý thông tin về các khóa học | Người dùng có thể thêm/sửa/xóa khóa học | 1 |
| PB13 | Quản lý lớp học | Admin / Giáo viên | Quản lý danh sách lớp học | Người dùng có thể theo dõi, cập nhật danh sách lớp học | 1 |
| PB14 | Quản lý người dùng | Admin | Quản lý tài khoản người dùng | Admin có thể thêm/sửa/xóa tài khoản người dùng | 1 |
| PB15 | Quản lý phân quyền | Admin | Phân quyền cho người dùng | Admin có thể cấp/quản lý quyền của từng tài khoản | 1 |
| PB16 | Quản lý phiếu chi | Admin / Nhân viên | Quản lý chi tiêu của trung tâm | Người dùng có thể tạo/sửa/xóa các phiếu chi | 2 |
| PB17 | Quản lý phiếu thu | Admin / Nhân viên | Quản lý thu nhập của trung tâm | Người dùng có thể tạo/sửa/xóa các phiếu thu | 2 |
| PB18 | Quản lý liên hệ | Admin / Nhân viên | Quản lý thông tin liên hệ của khách hàng | Người dùng có thể xem/xử lý các yêu cầu liên hệ | 2 |
| PB19 | Quản lý bài viết | Admin / Nhân viên | Quản lý nội dung tin tức và bài đăng | Người dùng có thể tạo/sửa/xóa bài viết | 2 |
| PB20 | Quản lý email | Admin / Nhân viên | Quản lý email gửi và nhận trong hệ thống | Người dùng có thể xem lịch sử email, gửi email hệ thống | 2 |
| PB21 | Xem thông tin khóa học | Khách hàng / Học viên | Xem thông tin về khóa học | Người dùng có thể tra cứu thông tin chi tiết của khóa học | 1 |
| PB22 | Thống kê báo cáo hệ thống | Admin / Nhân viên | Báo cáo và thống kê hệ thống | Người dùng có thể xem báo cáo hoạt động của trung tâm | 2 |
| PB23 | Ứng dụng AI trong hỗ trợ học tập | Học viên | Hỗ trợ học tập bằng AI | Học viên có thể nhận được sự hỗ trợ từ AI trong việc học | 2 |
| PB24 | Chatbox AI | Học viên / Khách hàng | Tương tác với chatbot AI | Người dùng có thể nhận tư vấn từ chatbot AI | 2 |
| PB25 | Quản lý Chatbox AI | Admin | Cấu hình và quản lý chatbot AI | Admin có thể thiết lập câu trả lời, cập nhật thông tin chatbot | 2 |

*Bảng 2.2: Bảng độ ưu tiên*

|  |  |  |
| --- | --- | --- |
| **Ký hiệu** | **Độ ưu tiên** | **Giải thích** |
| 1 | Cao | Cần thiết để sản phẩm hoạt động. |
| 2 | Trung bình | Quan trọng nhưng có thể làm sau khi hoàn thành các chức năng cần thiết. |
| 3 | Thấp | Có thể làm sau khi hoàn thành các chức năng cần thiết và quan trọng. |

* 1. **Mô tả Product Backlog**
     1. **PB01- Đăng kí**

|  |  |
| --- | --- |
| **ID** | **PB01** |
| **Tiêu đề** | Đăng ký |
| **Mô tả** | Khách vãng lai có thể đăng ký tài khoản truy cập Website |
| **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 vãng lai vào giao diện trang chủ, nhấn chọn nút “Đăng ký”. * Khi giao diện đăng ký đổ ra, tiếp tục nhập các thông tin yêu cầu vào form đăng ký. * Sau đó, nhấn nút “Tạo tài khoản” để gửi yêu cầu đăng ký tài khoản mới. * Hệ thống tiến hành kiểm tra email, mật khẩu. * Nếu email và mật khẩu hợp lệ, hệ thống hiển thị thông báo “Đăng ký thành công” và sẽ chuyển tới giao diện đăng nhập. * Nếu không hợp lệ, hệ thống sẽ hiển thị thông báo “Email đã tồn tại” hoặc “Mật khẩu chưa đúng định dạng”. |
| **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:** Spring Boot (xử lý xác thực) * **Database:** MySQL (lưu trữ tài khoản) * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Cao |

* + 1. **PB02 - Đăng nhập**

|  |  |
| --- | --- |
| **ID** | **PB02** |
| **Tiêu đề** | Đăng nhập |
| **Mô tả** | Người dùng có thể đăng nhập vào hệ thống bằng email và mật khẩu. |
| **Các bước thực hiện** | * Thiết kế giao diện đăng nhập. * Xử lý logic xác thực thông tin đăng nhập. * Kiểm tra thông tin tài khoản trong database. * Hiển thị thông báo đăng nhập thành công hoặc thất bại. * Điều hướng người dùng đến trang chính sau khi đăng nhập thành công. |
| **Điều kiện trước** | * Người dùng đã có tài khoản hợp lệ trong hệ thống. * Người dùng truy cập giao diện đăng nhập. * Nhập email và mật khẩu vào form đăng nhập. * Nhấn nút “Đăng nhập” để gửi yêu cầu xác thực. |
| **Điều kiện sau** | * Nếu đăng nhập thành công, người dùng sẽ vào hệ thống và sử dụng các chức năng phù hợp với vai trò của họ. * Nếu thất bại, hệ thống hiển thị thông báo lỗi “Email hoặc mật khẩu không đúng”. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện đăng nhập) * **Backend:** Spring Boot (xử lý xác thực) * **Database:** MySQL (lưu trữ tài khoản) * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Cao |

* + 1. **PB03 - Đăng xuất**

|  |  |
| --- | --- |
| **ID** | **PB03** |
| **Tiêu đề** | Đăng xuất |
| **Mô tả** | Người dùng có thể đăng xuất khỏi hệ thống. |
| **Các bước thực hiện** | * Thiết kế giao diện đăng xuất: * Thêm nút "Đăng xuất" vào giao diện (thanh menu hoặc trang cài đặt tài khoản). * Đảm bảo vị trí của nút thuận tiện để người dùng dễ thao tác. * Xử lý logic đăng xuất: * Khi người dùng nhấn nút "Đăng xuất", hiển thị hộp thoại xác nhận "Bạn có chắc chắn muốn đăng xuất không?". * Nếu người dùng xác nhận, gửi yêu cầu đăng xuất đến backend. * Hủy token xác thực: * Nếu sử dụng JWT Token, backend sẽ yêu cầu frontend xóa token khỏi Local Storage hoặc Cookies. * Nếu sử dụng Session, backend sẽ hủy session đang hoạt động của người dùng. * Cập nhật trạng thái người dùng: * Backend cập nhật trạng thái của người dùng trong cơ sở dữ liệu nếu cần (ví dụ: đánh dấu trạng thái "offline"). * Điều hướng người dùng: * Sau khi đăng xuất thành công, hệ thống điều hướng người dùng đến trang đăng nhập hoặc trang chủ. * Nếu đăng xuất thất bại do lỗi hệ thống, hiển thị thông báo "Đăng xuất không thành công, vui lòng thử lại". |
| **Điều kiện trước** | * Người dùng đã đăng nhập vào hệ thống. * Người dùng truy cập giao diện chính và nhấn chọn nút “Đăng xuất”. * Xác nhận thao tác đăng xuất. |
| **Điều kiện sau** | * Người dùng được đăng xuất hoàn toàn khỏi hệ thống. * Phiên làm việc (session/token) bị hủy, ngăn chặn truy cập trái phép. * Nếu đăng xuất thành công, người dùng được chuyển hướng đến trang đăng nhập hoặc trang chủ. * Nếu xảy ra lỗi trong quá trình đăng xuất, hệ thống hiển thị thông báo lỗi. |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB04 – Quên mật khẩu**

|  |  |
| --- | --- |
| **ID** | **PB04** |
| **Tiêu đề** | Quên mật khẩu |
| **Mô tả** | Người dùng có thể đặt lại mật khẩu khi quên. |
| **Các bước thực hiện** | * Thiết kế giao diện quên mật khẩu với trường nhập email. * Hệ thống kiểm tra email nhập vào: * Nếu email không tồn tại trong hệ thống, hiển thị thông báo lỗi. * Nếu email hợp lệ, hệ thống gửi mã OTP qua email. * Người dùng nhập mã OTP và mật khẩu mới. * Hệ thống xác thực mã OTP: * Nếu hợp lệ, cập nhật mật khẩu mới vào hệ thống. * Nếu không hợp lệ, hiển thị thông báo lỗi. * Hiển thị thông báo thành công hoặc thất bại và điều hướng đến trang đăng nhập nếu thành công. |
| **Điều kiện trước** | * Người dùng vào giao diện đăng nhập, nhấn chọn “Quên mật khẩu”. * Nhập email đã đăng ký và gửi yêu cầu đặt lại mật khẩu. * Hệ thống kiểm tra email hợp lệ và gửi email chứa liên kết đặt lại mật khẩu. * Người dùng truy cập liên kết và đặt mật khẩu mới. |
| **Điều kiện sau** | * Người dùng có thể sử dụng mật khẩu mới để đăng nhập. * Hệ thống hiển thị thông báo “Đổi mật khẩu thành công” |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * Database: MySQL * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB05 Thay đổi mật khẩu**

|  |  |
| --- | --- |
| **ID** | PB05 |
| **Tiêu đề** | Thay đổi mật khẩu |
| **Mô tả** | Người dùng có thể thay đổi mật khẩu của mình trên hệ thống. |
| **Các bước thực hiện** | * Thiết kế giao diện đổi mật khẩu với các trường nhập: mật khẩu cũ, mật khẩu mới, xác nhận mật khẩu mới. * Người dùng nhập thông tin và gửi yêu cầu. * Hệ thống kiểm tra mật khẩu cũ: * Nếu mật khẩu cũ đúng, tiếp tục kiểm tra mật khẩu mới. * Nếu sai, hiển thị thông báo lỗi. * Xác thực mật khẩu mới: * Nếu hợp lệ theo yêu cầu bảo mật, cập nhật trong hệ thống. * Nếu không hợp lệ, hiển thị thông báo lỗi. * Hiển thị thông báo thành công hoặc thất bại. |
| **Điều kiện trước** | * Tài khoản người dùng phải tồn tại trong hệ thống. |
| **Điều kiện sau** | * Mật khẩu người dùng được thay đổi thành công. |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Springboot * Database: MySQL * API: RESTful API |
| **Độ phức tạp** | Trung bình |

* + 1. **PB06: Quản lý giáo viên**

|  |  |
| --- | --- |
| **ID** | **PB01** |
| **Tiêu đề** | Quản lý giáo viên |
| **Mô tả** | Hệ thống cho phép quản lý thông tin giáo viên bao gồm thêm, sửa, xóa và xem danh sách giáo viên. |
| **Các bước thực hiện** | * Thiết kế giao diện danh sách giáo viên với chức năng tìm kiếm, lọc và phân trang. * Cho phép tạo mới giáo viên bằng cách nhập các thông tin cơ bản. * Cho phép chỉnh sửa thông tin giáo viên. * Cho phép xóa giáo viên khỏi hệ thống. * Lưu thông tin vào cơ sở dữ liệu. |
| **Điều kiện trước** | * Người dùng có quyền quản trị viên. * Truy cập trang quản lý giáo viên. * Chọn thao tác cần thực hiện (thêm, sửa, xóa). |
| **Điều kiện sau** | * Nếu thêm mới, hệ thống hiển thị "Thêm giáo viên thành công". * Nếu chỉnh sửa, hệ thống cập nhật thông tin và hiển thị "Cập nhật thành công". * Nếu xóa, hệ thống xác nhận trước khi xóa và hiển thị "Xóa giáo viên thành công". |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * Database: MySQL * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB07 - Quản lý nhân viên**

|  |  |
| --- | --- |
| **ID** | **PB01** |
| **Tiêu đề** | Quản lý nhân viên |
| **Mô tả** | Khách vãng lai có thể đăng ký tài khoản truy cập Website |
| **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 vãng lai vào giao diện trang chủ, nhấn chọn nút “Đăng ký”. * Khi giao diện đăng ký đổ ra, tiếp tục nhập các thông tin yêu cầu vào form đăng ký. * Sau đó, nhấn nút “Tạo tài khoản” để gửi yêu cầu đăng ký tài khoản mới. * Hệ thống tiến hành kiểm tra email, mật khẩu. * Nếu email và mật khẩu hợp lệ, hệ thống hiển thị thông báo “Đăng ký thành công” và sẽ chuyển tới giao diện đăng nhập. * - Nếu không hợp lệ, hệ thống sẽ hiển thị thông báo “Email đã tồn tại” hoặc “Mật khẩu chưa đúng định dạng”. |
| **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 nhập) * Backend: Spring Boot (xử lý xác thực) * Database: MySQL (lưu trữ tài khoản) * API: RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Cao |

* + 1. **PB08 - Quản lý học viên**

|  |  |
| --- | --- |
| **ID** | **PB01** |
| **Tiêu đề** | Quản lý học viên |
| **Mô tả** | Hệ thống hỗ trợ quản lý danh sách học viên, bao gồm thêm mới, chỉnh sửa, xóa và xem thông tin chi tiết. |
| **Các bước thực hiện** | * Quản trị viên hoặc giáo viên truy cập trang quản lý học viên. * Thực hiện thao tác thêm, sửa hoặc xóa học viên theo nhu cầu. * Hệ thống kiểm tra và lưu thông tin học viên vào database. * Hiển thị danh sách học viên đã cập nhật. |
| **Điều kiện trước** | * Người dùng có quyền truy cập quản lý học viên. * Truy cập trang quản lý học viên. * Chọn thao tác cần thực hiện (thêm, sửa, xóa). |
| **Điều kiện sau** | * Nếu thêm mới, hệ thống hiển thị "Thêm học viên thành công". * Nếu chỉnh sửa, hệ thống cập nhật thông tin và hiển thị "Cập nhật thành công". * Nếu xóa, hệ thống xác nhận trước khi xóa và hiển thị "Xóa học viên thành công". |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * Database: MySQL * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB09 - Quản lý chi tiết lớp học**

|  |  |
| --- | --- |
| **ID** | **PB09** |
| **Tiêu đề** | Quản lý chi tiết lớp học |
| **Mô tả** | Hệ thống cho phép quản trị viên tạo, cập nhật, xóa và xem danh sách lớp học. |
| **Các bước thực hiện** | * Thiết kế giao diện quản lý lớp học. * Xử lý logic thêm, sửa, xóa, xem danh sách lớp học. * Liên kết lớp học với khóa học và giáo viên giảng dạy. * Lưu trữ dữ liệu lớp học vào database. * Hiển thị danh sách lớp học trên giao diện. |
| **Điều kiện trước** | * Quản trị viên có quyền truy cập chức năng quản lý lớp học. * Dữ liệu khóa học và giáo viên đã có trong hệ thống. * Người dùng truy cập giao diện quản lý lớp học. |
| **Điều kiện sau** | * Nếu thao tác thành công, hệ thống cập nhật danh sách lớp học vào database. * Nếu thao tác thất bại, hiển thị thông báo lỗi tương ứng. |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * Database: MySQL * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB010 - Quản lý lịch thi**

|  |  |
| --- | --- |
| **ID** | **PB01** |
| **Tiêu đề** | Quản lý lịch thi |
| **Mô tả** | Hệ thống cho phép quản trị viên tạo, chỉnh sửa, xóa và quản lý lịch thi. |
| **Các bước thực hiện** | * Thiết kế giao diện quản lý lịch thi. * Xử lý logic tạo, sửa, xóa lịch thi. * Gửi thông báo lịch thi đến học viên và giáo viên. * Lưu trữ dữ liệu lịch thi vào database. |
| **Điều kiện trước** | * Quản trị viên có quyền truy cập chức năng quản lý lịch thi. * Danh sách khóa học và lớp học đã có trong hệ thống. * Người dùng truy cập giao diện quản lý lịch thi. |
| **Điều kiện sau** | * Nếu thao tác thành công, hệ thống cập nhật danh sách lịch thi vào database. * Nếu thao tác thất bại, hiển thị thông báo lỗi tương ứng. |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * Database: MySQL * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB011 - Quản lý chức vụ**

|  |  |
| --- | --- |
| **ID** | **PB011** |
| **Tiêu đề** | Quản lý chức vụ |
| **Mô tả** | Hệ thống cho phép quản trị viên tạo, cập nhật, xóa và xem danh sách chức vụ trong hệ thống. |
| **Các bước thực hiện** | * Thiết kế giao diện quản lý chức vụ. * Xử lý logic thêm, sửa, xóa, xem danh sách chức vụ. * Liên kết chức vụ với tài khoản người dùng. * Lưu trữ dữ liệu chức vụ vào database. * Hiển thị danh sách chức vụ trên giao diện. |
| **Điều kiện trước** | * Quản trị viên có quyền truy cập chức năng quản lý chức vụ. * Dữ liệu tài khoản người dùng đã có trong hệ thống. * Người dùng truy cập giao diện quản lý chức vụ. |
| **Điều kiện sau** | * Nếu thao tác thành công, hệ thống cập nhật danh sách chức vụ vào database. * Nếu thao tác thất bại, hiển thị thông báo lỗi tương ứng. |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * Database: MySQL * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB012- Quản lý khóa học**

|  |  |
| --- | --- |
| **ID** | **PB010** |
| **Tiêu đề** | Quản lý khóa học |
| **Mô tả** | Hệ thống cho phép quản trị viên tạo, cập nhật, xóa và xem danh sách khóa học. |
| **Các bước thực hiện** | * Thiết kế giao diện quản lý khóa học. * Xử lý logic thêm, sửa, xóa, xem danh sách khóa học. * Liên kết khóa học với giáo viên giảng dạy và lớp học. * Lưu trữ dữ liệu khóa học vào database. * Hiển thị danh sách khóa học trên giao diện. |
| **Điều kiện trước** | * Quản trị viên có quyền truy cập chức năng quản lý khóa học. * Dữ liệu lớp học và giáo viên đã có trong hệ thống. * Người dùng truy cập giao diện quản lý khóa học. |
| **Điều kiện sau** | * Nếu thao tác thành công, hệ thống cập nhật danh sách khóa học vào database. * Nếu thao tác thất bại, hiển thị thông báo lỗi tương ứng. |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * Database: MySQL * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB13 - Quản lý lớp học**

|  |  |
| --- | --- |
| **ID** | **PB011** |
| **Tiêu đề** | Quản lý lớp học |
| **Mô tả** | Hệ thống cho phép quản trị viên tạo, cập nhật, xóa và xem danh sách lớp học. |
| **Các bước thực hiện** | * Thiết kế giao diện quản lý lớp học. * Xử lý logic thêm, sửa, xóa, xem danh sách lớp học. * Liên kết lớp học với khóa học và giáo viên giảng dạy. * Lưu trữ dữ liệu lớp học vào database. * Hiển thị danh sách lớp học trên giao diện. |
| **Điều kiện trước** | * Quản trị viên có quyền truy cập chức năng quản lý lớp học. * Dữ liệu khóa học và giáo viên đã có trong hệ thống. * Người dùng truy cập giao diện quản lý lớp học. * Hệ thống tiến hành kiểm tra email, mật khẩu. |
| **Điều kiện sau** | * Nếu thao tác thành công, hệ thống cập nhật danh sách lớp học vào database. * Nếu thao tác thất bại, hiển thị thông báo lỗi tương ứng. |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * Database: MySQL * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB14 - Quản lý người dùng**

|  |  |
| --- | --- |
| **ID** | **PB014** |
| **Tiêu đề** | Quản lý người dùng |
| **Mô tả** | Hệ thống cho phép quản trị viên tạo, cập nhật, xóa và xem danh sách người dùng trong hệ thống. |
| **Các bước thực hiện** | * Thiết kế giao diện quản lý người dùng. * Xử lý logic thêm, sửa, xóa, xem danh sách người dùng. * Phân quyền và gán vai trò cho từng tài khoản. * Lưu trữ dữ liệu người dùng vào database. * Hiển thị danh sách người dùng trên giao diện. |
| **Điều kiện trước** | * Quản trị viên có quyền truy cập chức năng quản lý người dùng. * Dữ liệu tài khoản đã có trong hệ thống. * Người dùng truy cập giao diện quản lý người dùng. |
| **Điều kiện sau** | * Nếu thao tác thành công, hệ thống cập nhật danh sách người dùng vào database. * Nếu thao tác thất bại, hiển thị thông báo lỗi tương ứng |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * Database: MySQL * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB15- Phân quyền người dùng**

|  |  |
| --- | --- |
| **ID** | **PB015** |
| **Tiêu đề** | Phân quyền người dùng |
| **Mô tả** | Hệ thống cho phép quản trị viên cấp quyền truy cập theo vai trò cho từng tài khoản. |
| **Các bước thực hiện** | * Thiết kế giao diện quản lý phân quyền. * Xử lý logic gán quyền, chỉnh sửa và thu hồi quyền truy cập. * Liên kết vai trò với quyền truy cập phù hợp. * Lưu trữ dữ liệu phân quyền vào database. * Hiển thị danh sách phân quyền trên giao diện. |
| **Điều kiện trước** | * Quản trị viên có quyền truy cập chức năng quản lý phân quyền. * Dữ liệu tài khoản và vai trò đã có trong hệ thống. |
| **Điều kiện sau** | * Nếu thao tác thành công, hệ thống cập nhật danh sách người dùng vào database. * Nếu thao tác thất bại, hiển thị thông báo lỗi tương ứng. |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * Database: MySQL * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB16- Quản lý phiếu chi**

|  |  |
| --- | --- |
| **ID** | **PB016** |
| **Tiêu đề** | Quản lý phiếu chi |
| **Mô tả** | Hệ thống cho phép kế toán tạo, cập nhật, xóa và xem danh sách phiếu chi. |
| **Các bước thực hiện** | * Thiết kế giao diện quản lý phiếu chi. * Xử lý logic thêm, sửa, xóa phiếu chi. * Ghi nhận thông tin thanh toán và lưu trữ dữ liệu phiếu chi vào database. * Hiển thị danh sách phiếu chi trên giao diện. |
| **Điều kiện trước** | * Kế toán viên có quyền truy cập chức năng quản lý phiếu chi. * Dữ liệu tài chính đã có trong hệ thống. |
| **Điều kiện sau** | * Nếu thao tác thành công, hệ thống cập nhật danh sách phiếu chi vào database. * Nếu thao tác thất bại, hiển thị thông báo lỗi tương ứng |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * Database: MySQL * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB17 - Quản lý phiếu thu**

|  |  |
| --- | --- |
| **ID** | **PB017** |
| **Tiêu đề** | Quản lý phiếu thu |
| **Mô tả** | Hệ thống cho phép kế toán viên tạo, chỉnh sửa, xóa và quản lý phiếu thu trong hệ thống. |
| **Các bước thực hiện** | * Thiết kế giao diện quản lý phiếu thu. * Xử lý logic thêm, sửa, xóa phiếu thu. * Ghi nhận thông tin thanh toán và lưu trữ dữ liệu phiếu thu vào database. * Hiển thị danh sách phiếu thu trên giao diện. |
| **Điều kiện trước** | * Kế toán viên có quyền truy cập chức năng quản lý phiếu thu. * Dữ liệu tài chính đã có trong hệ thống. |
| **Điều kiện sau** | * Nếu thao tác thành công, hệ thống cập nhật danh sách phiếu thu vào database. * Nếu thao tác thất bại, hiển thị thông báo lỗi tương ứng. |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * Database: MySQL * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB18- Quản lý liên hệ**

|  |  |
| --- | --- |
| **ID** | **PB018** |
| **Tiêu đề** | Quản lý liên hệ |
| **Mô tả** | Hệ thống cho phép quản trị viên quản lý danh sách liên hệ của khách hàng. |
| **Các bước thực hiện** | * Thiết kế giao diện quản lý liên hệ. * Xử lý logic thêm, sửa, xóa liên hệ. * Lưu trữ thông tin liên hệ vào database. * Hiển thị danh sách liên hệ trên giao diện. |
| **Điều kiện trước** | * Người dùng có quyền truy cập chức năng quản lý liên hệ. * Danh sách liên hệ đã có trong hệ thống. |
| **Điều kiện sau** | * Nếu thao tác thành công, hệ thống cập nhật danh sách liên hệ vào database. * Nếu thao tác thất bại, hiển thị thông báo lỗi tương ứng. |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * Database: MySQL * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB19 - Quản lý bài viết**

|  |  |
| --- | --- |
| **ID** | **PB019** |
| **Tiêu đề** | Quản lý bài viết |
| **Mô tả** | Hệ thống cho phép quản trị viên quản lý nội dung bài viết trên nền tảng. |
| **Các bước thực hiện** | * Thiết kế giao diện quản lý bài viết. * Xử lý logic thêm, sửa, xóa bài viết. * Lưu trữ nội dung bài viết vào database. * Hiển thị danh sách bài viết trên giao diện. |
| **Điều kiện trước** | * Người dùng có quyền truy cập chức năng quản lý bài viết. * Danh sách bài viết đã có trong hệ thống. |
| **Điều kiện sau** | * Nếu thao tác thành công, hệ thống cập nhật danh sách bài viết vào database. * Nếu thao tác thất bại, hiển thị thông báo lỗi tương ứng. |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * Database: MySQL * API: RESTful API. |
| **Độ phức tạp** | Cao |

* + 1. **PB120 - Quản lý email**

|  |  |
| --- | --- |
| **ID** | **PB020** |
| **Tiêu đề** | Quản lý email |
| **Mô tả** | Hệ thống hỗ trợ quản trị viên gửi và nhận email từ người dùng. |
| **Các bước thực hiện** | * Thiết kế giao diện gửi và quản lý email. * Tích hợp API gửi và nhận email. * Lưu trữ thông tin email vào database. * Hiển thị danh sách email trên giao diện. |
| **Điều kiện trước** | * Người dùng có quyền sử dụng chức năng quản lý email. |
| **Điều kiện sau** | * Nếu thao tác thành công, email được gửi hoặc lưu trữ. * Nếu thao tác thất bại, hiển thị thông báo lỗi tương ứng. |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * Database: MySQL * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB21 Xem thông tin khóa học**

|  |  |
| --- | --- |
| **ID** | **PB021** |
| **Tiêu đề** | Xem thông tin khóa học |
| **Mô tả** | Người dùng có thể xem danh sách khóa học, bao gồm thông tin chi tiết như tên khóa học, mô tả, giảng viên, thời gian học và giá tiền. |
| **Các bước thực hiện** | * Thiết kế giao diện danh sách khóa học hiển thị thông tin cơ bản. * Xây dựng API để lấy dữ liệu khóa học từ cơ sở dữ liệu. * Tích hợp API vào giao diện để hiển thị dữ liệu khóa học. * Cho phép người dùng lọc và tìm kiếm khóa học theo tiêu chí như giảng viên, chủ đề, giá cả. * Hiển thị chi tiết khóa học khi người dùng nhấp vào một khóa học cụ thể. |
| **Điều kiện trước** | * Người dùng đã đăng nhập vào hệ thống hoặc truy cập với tư cách khách vãng lai. |
| **Điều kiện sau** | * Giao diện hiển thị danh sách khóa học và thông tin chi tiết khi người dùng chọn một khóa học. * Người dùng có thể tìm kiếm, lọc và xem khóa học theo nhu cầu |
| **Công nghệ sử dụng** | * Frontend: React.js * Backend: Spring Boot * Database: MySQL * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB22 - Thống kê báo cáo hệ thống**

|  |  |
| --- | --- |
| **ID** | **PB022** |
| **Tiêu đề** | Thống kê báo cáo |
| **Mô tả** | * Quản trị viên có thể xem các thống kê liên quan đến số lượng người dùng, khóa học, giao dịch và các hoạt động khác trong hệ thống. |
| **Các bước thực hiện** | * Xây dựng giao diện báo cáo thống kê với các biểu đồ trực quan. * Xây dựng API để truy xuất dữ liệu từ cơ sở dữ liệu. * Thiết kế hệ thống lọc dữ liệu theo thời gian, loại báo cáo. * Hiển thị dữ liệu thống kê theo thời gian thực. * Cho phép xuất báo cáo dưới dạng PDF hoặc Excel. |
| **Điều kiện trước** | * Quản trị viên có thể xem báo cáo thống kê với dữ liệu chính xác. * Có thể tải xuống báo cáo dưới các định dạng khác nhau. |
| **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 (hiển thị báo cáo). * Backend: Spring Boot (truy xuất dữ liệu thống kê). * Database: MySQL (lưu trữ dữ liệu). * Thư viện hỗ trợ: Chart.js (hiển thị biểu đồ), Apache POI (xuất Excel), iText (xuất PDF). |
| **Độ phức tạp** | Cao |

* + 1. **PB23- Ứng dụng AI trong hỗ trợ học tập**

|  |  |
| --- | --- |
| **ID** | **PB023** |
| **Tiêu đề** | Ứng dụng AI trong học tập |
| **Mô tả** | Hệ thống sử dụng AI để cung cấp gợi ý học tập, đánh giá tiến độ học tập và hỗ trợ giải đáp thắc mắc của học viên. |
| **Các bước thực hiện** | * Phát triển AI để phân tích dữ liệu học tập của học viên. * Xây dựng hệ thống gợi ý bài học phù hợp với từng học viên. * Tích hợp chatbot AI để hỗ trợ giải đáp thắc mắc. * Hiển thị báo cáo tiến độ học tập cho học viên và giảng viên. |
| **Điều kiện trước** | * Học viên đã đăng ký tài khoản và có dữ liệu học tập trong hệ thống. |
| **Điều kiện sau** | * Hệ thống AI có thể phân tích dữ liệu và đưa ra gợi ý học tập chính xác. * Chatbot AI có thể hỗ trợ học viên giải đáp thắc mắc nhanh chóng. |
| **Công nghệ sử dụng** | * AI & Machine Learning: Python (TensorFlow, scikit-learn). * Frontend: React.js. * Backend: Spring Boot. * Database: MySQL. * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB024 -Chatbot AI**

|  |  |
| --- | --- |
| **ID** | **PB024** |
| **Tiêu đề** | Chatbox AI |
| **Mô tả** | Học viên có thể trò chuyện với AI chatbot để nhận hỗ trợ về học tập, thông tin khóa học, và hướng dẫn sử dụng hệ thống. |
| **Các bước thực hiện** | * Phát triển AI chatbot với khả năng xử lý ngôn ngữ tự nhiên (NLP). * Tích hợp chatbot vào giao diện website. * Xây dựng API để chatbot có thể truy xuất dữ liệu từ hệ thống. * Tạo giao diện trò chuyện thân thiện với người dùng. |
| **Điều kiện trước** | * Học viên đã đăng nhập vào hệ thống. |
| **Điều kiện sau** | * Học viên có thể tương tác với chatbot để nhận hỗ trợ nhanh chóng. * Chatbot có thể trả lời các câu hỏi một cách chính xác. |
| **Công nghệ sử dụng** | * AI & NLP: Dialogflow, OpenAI GPT. * Frontend: React.js. * Backend: Node.js. * Database: MySQL. * API: RESTful API |
| **Độ phức tạp** | Cao |

* + 1. **PB25-Quản lý Chatbot AI**

|  |  |
| --- | --- |
| **ID** | **PB025** |
| **Tiêu đề** | Quản lý chatbox AI |
| **Mô tả** | Quản trị viên có thể quản lý dữ liệu huấn luyện AI chatbot, theo dõi hiệu suất chatbot và cải thiện khả năng phản hồi. |
| **Các bước thực hiện** | * Xây dựng giao diện quản trị chatbot AI. * Cho phép quản trị viên chỉnh sửa dữ liệu huấn luyện chatbot. * Theo dõi lịch sử trò chuyện và đánh giá phản hồi của chatbot. * Tích hợp công cụ phân tích hiệu suất chatbot. |
| **Điều kiện trước** | * Quản trị viên đã đăng nhập vào hệ thống. |
| **Điều kiện sau** | * Quản trị viên có thể cập nhật dữ liệu và cải thiện chatbot. * Chatbot hoạt động tốt hơn theo thời gian. |
| **Công nghệ sử dụng** | * AI & NLP: Dialogflow, OpenAI GPT. * Frontend: React.js. * Backend: Node.js. * Database: MySQL. * API: RESTful API. |
| **Độ phức tạp** | Cao |

## 2.3. Chia theo từng Sprirnt

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint** | **Chức năng (Function)** | **Ngày bắt đầu (Started Date)** | **Ngày kết thúc (Due Date)** |
| Sprint 1 | PB01, PB02, PB03, PB04, PB05, PB06, PB07, PB08, PB09, PB13, PB14, PB15, PB16, PB17 | 31/03/2025 | 18/04/2025 |
| Sprint 2 | PB10, PB11, PB12, PB18, PB19, PB20, PB21, PB22, PB23, PB24, PB25 | 19/04/2025 | 12/05/2025 |

1. **HẠN CHẾ**

|  |  |
| --- | --- |
| Hạn chế | Tình trạng |
| Thời gian | Thời gian để hoàn thành dự án là 639 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. |