**ĐẠI HỌC DUY TÂN**

**TRƯỜNG KHMT**

**KHOA CÔNG NGHỆ THÔNG TIN**

**--------------🙖🙐✰🙖🙐-------------**

****

**Tên đề tài:**

**XÂY DỰNG WEBSITE HOMITUTOR – NỀN TẢNG**

**GIA SƯ THÔNG MINH VỚI AI**

**(TÀI LIỆU PRODUCT BACKLOG)**

**GVHD:** Th. S Hồ Lê Viết Nin

**Thành Viên**

Nguyễn Thu Hạ Nhi 27217536629

Trần Hữu Lợi 27211236628

Đặng Văn Mạnh 27214303104

Đỗ Ngọc Hiển 27211254139

Trần Đình Nhật 27211235104

**Đà Nẵng, Tháng 03 năm 2025**

**THÔNG TIN DỰ ÁN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dự án viết tắt** | **HOMITUTOR** | | | |
| **Tên dự án** | Xây dựng website HomiTutor - Nền tảng gia sư thông minh với 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** | Nguyễn Thu Hạ Nhi | nnhi85788@gmail.com | | 0834088540 |
| **Thành viên nhóm** | Trần Hữu Lợi | tranhuuloi2k3@gmail.com | | 0328710340 |
| Đặng Văn Mạnh | danhvanmanh28082002@gmail.com | | 0898776838 |
| Đỗ Ngọc Hiển | ngok13051211@gmail.com | | 0379000134 |
| Trần Đình Nhật | nhat01012003@gmail.com | | 0396069736 |

|  |  |  |  |
| --- | --- | --- | --- |
| **TÊN TÀI LIỆU** | | | |
| **Tên dự án** | Xây dựng website HomiTutor - Nền tảng gia sư thông minh với AI | | |
| **Tiêu đề tài liệu** | Product Backlog | | |
| **Người thực hiện** | Nguyễn Thu Hạ Nhi | | |
| **Ngày** | 21/03/2025 | **Tên tệp:** | [KLTN-45] Proposal Nhom.docx |
| **URL** |  | | |
| **Access** | Khoa CNTT | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **LỊCH SỬ BẢN SỬA** | | | |
| **Version** | **Person** | **Date** | **Description** |
| 1.0 | Nguyễn Thu Hạ Nhi | 20/03/2025 | Tạo tài liệu |
| 1.1 | Nguyễn Thu Hạ Nhi | 22/03/2025 | 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 | **Signature:** |  |
| **Date:** | 22/03/2025 |
| **Chủ sở hữu** | | Hồ Lê Viết Nin | **Signature:** |  |
| **Date:** | 22/03/2025 |
| **Đội sản xuất** | | Nguyễn Thu Hạ Nhi | **Signature:** |  |
| **Date:** | 22/03/2025 |
| **Thành viên** | | Trần Hữu Lợi | **Signature:** |  |
| **Date:** | 22/03/2025 |
| Đặng Văn Mạnh | **Signature:** |  |
| **Date:** | 22/03/2025 |
| Đỗ Ngọc Hiển | **Signature:** |  |
| **Date:** | 22/03/2025 |
| Trần Đình Nhật | **Signature:** |  |
| **Date:** | 22/03/2025 |

**MỤC LỤC**

[**1. GIỚI THIỆU 1**](#_Toc193723954)

[**1.1. Mục đích 1**](#_Toc193723955)

[**Tài liệu này xác định danh sách các yêu cầu cần thiết cho hệ thống HomiTutor - Nền tảng gia sư thông minh với AI. Product Backlog giúp theo dõi các tính năng cần phát triển theo ưu tiên, phục vụ việc lập kế hoạch và triển khai dự án theo mô hình Scrum. 1**](#_Toc193723956)

[**HomiTutor hướng đến việc xây dựng một nền tảng kết nối học viên với gia sư một cách thông minh, tiện lợi và hiệu quả. Hệ thống hỗ trợ việc tìm kiếm, đặt lịch, quản lý lớp học, thanh toán và hỗ trợ học tập thông qua AI, giúp học viên tiếp cận được giáo viên phù hợp nhất với nhu cầu của họ. 1**](#_Toc193723957)

[**1.2. Phạm vi 1**](#_Toc193723958)

[**1.3. Tham khảo 1**](#_Toc193723959)

[**1.4. Biểu đồ UseCase Tổng quát 2**](#_Toc193723960)

[**2. Product Backlog 3**](#_Toc193723961)

[**2.1. Danh sách các Product Backlog trong Hệ thống 3**](#_Toc193723964)

[**2.1.1 PB01-Đăng ký tài khoản 4**](#_Toc193723965)

[**2.1.2 PB02-Đăng nhập 5**](#_Toc193723966)

[**2.1.3 PB05-Tìm kiếm gia sư 6**](#_Toc193723967)

[**2.1.4 PB07-Đặt lịch học 7**](#_Toc193723969)

[**2.1.5 PB10-Thanh toán online 8**](#_Toc193723971)

[**2.1.6 PB06-Xem chi tiết gia sư 9**](#_Toc193723972)

[**2.1.7 PB08 – Tra cứu lịch học 10**](#_Toc193723973)

[**2.1.8 PB04-Quản lý thông tin cá nhân 11**](#_Toc193723974)

[**2.1.9 PB09-Quản lý lịch dạy (Gia sư) 12**](#_Toc193723975)

[**2.1.10 PB13-Hỗ trợ Chatbot AI 13**](#_Toc193723976)

[**2.1.11 PB03-Quên mật khẩu 14**](#_Toc193723977)

[**2.1.12 PB11-Đánh giá gia sư 15**](#_Toc193723978)

[**2.1.13 PB12-Quản lý thanh toán (Admin) 15**](#_Toc193723979)

[**2.1.14 PB14-Quản lý tài khoản (Admin) 16**](#_Toc193723981)

[**2.1.15 PB15-Quản lý báo cáo và phân tích dữ liệu 17**](#_Toc193723982)

[**2.1.16 PB16 – Quản lý kho tài liệu phản hồi 18**](#_Toc193723983)

[**2. Phân tích về Sprint Backlog 20**](#_Toc193723984)

[**3. Hạn chế 20**](#_Toc193723985)

[**4. Tóm lược 20**](#_Toc193723986)

# 

# GIỚI THIỆU

## Mục đích

## Tài liệu này xác định danh sách các yêu cầu cần thiết cho hệ thống **HomiTutor - Nền tảng gia sư thông minh với AI**. Product Backlog giúp theo dõi các tính năng cần phát triển theo ưu tiên, phục vụ việc lập kế hoạch và triển khai dự án theo mô hình Scrum.

## HomiTutor hướng đến việc xây dựng một nền tảng kết nối học viên với gia sư một cách thông minh, tiện lợi và hiệu quả. Hệ thống hỗ trợ việc tìm kiếm, đặt lịch, quản lý lớp học, thanh toán và hỗ trợ học tập thông qua AI, giúp học viên tiếp cận được giáo viên phù hợp nhất với nhu cầu của họ.

## Phạm vi

Bên cạnh việc cung cấp cho bạn các tài liệu, nó sẽ liệt kê tất cả các thành viên trong nhóm phát triển trong dự án này, quy trình và khuôn khổ mà dự án sẽ áp dụng.

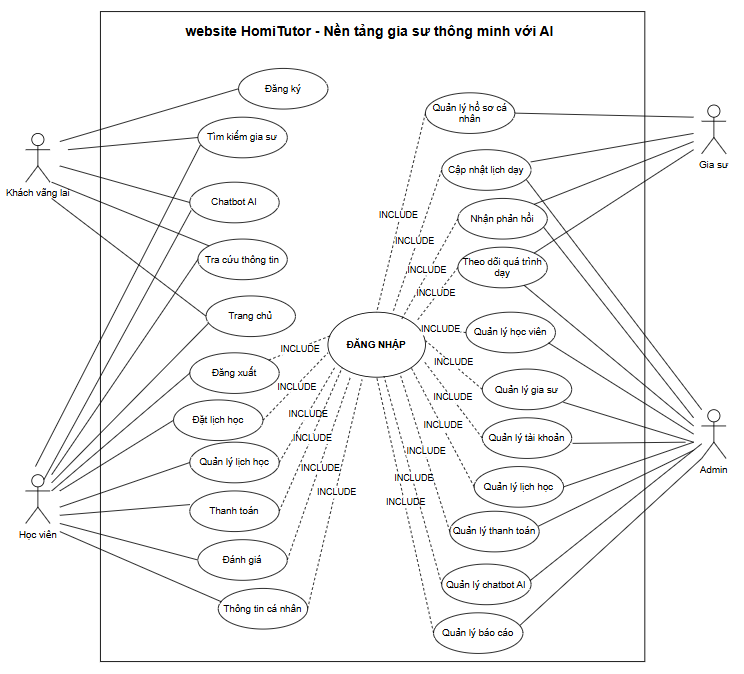
Tài liệu này đưa ra kế hoạch cho từng giai đoạn của quy trình phát triển phần mềm dự trên quy trình Scrum bao gồm: thời gian bắt đầu, thời gian kết thúc và một số ngày làm việc. Đây là kế hoạch chung và sẽ được cập nhật chi tiết về quy trình phát triển phần mềm trong phiên bản tiếp theo của tài liệu. Đề xuất bao gồm giới thiệu các giải pháp, xác định cách tốt nhất để phát triển mà chúng tôi tạo ra, tổng chi phí ước tính, thời gian hoàn vốn và khối lượng hòa vốn cho dự án.

## 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 | NextJS | https://nextjs.org/docs/pages/building-your-application/authentication |
| 5 | NestJS | https://docs.nestjs.com/ |
| 6 | NodeJS | https://www.nodejs.org/en/learn/getting-started/introduction-to-nodejs |
| 7 | 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 tài khoản | Học viên/Gia sư | Tạo tài khoản trên hệ thống | Có thể đăng ký với email, số điện thoại | 1 |
| PB02 | Đăng nhập | Học viên/Gia sư | Truy cập vào hệ thống | Có thể đăng nhập bằng email/mật khẩu | 1 |
| PB03 | Quên mật khẩu | Học viên/Gia sư | Lấy lại mật khẩu | Hệ thống gửi email khôi phục mật khẩu | 2 |
| PB04 | Quản lý thông tin cá nhân | Học viên/Gia sư | Chỉnh sửa hồ sơ cá nhân | Có thể cập nhật thông tin như tên, địa chỉ, ảnh đại diện | 3 |
| PB05 | Tìm kiếm gia sư | Học viên | Tìm kiếm gia sư theo môn học | Có thể lọc theo đánh giá, địa điểm | 1 |
| PB06 | Xem chi tiết gia sư | Học viên | Xem thông tin về gia sư | Hiển thị thông tin chi tiết của gia sư | 1 |
| PB07 | Đặt lịch học | Học viên | Đặt lịch với gia sư | Có thể chọn thời gian, xác nhận với gia sư | 1 |
| PB08 | Tra cứu lịch học | Học viên | Xem lịch học | Hiển thị danh sách lịch học đã đặt | 2 |
| PB09 | Quản lý lịch dạy | Gia sư | Xem và quản lý lịch dạy | Hiển thị danh sách lịch dạy | 2 |
| PB10 | Thanh toán online | Học viên | Thanh toán online | Hỗ trợ các phương thức thanh toán phổ biến | 1 |
| PB11 | Đánh giá gia sư | Học viên | Đánh giá chất lượng gia sư | Cho phép đánh giá bằng sao và bình luận | 3 |
| PB12 | Quản lý thanh toán | Admin | Kiểm soát giao dịch | Xem danh sách thanh toán, xử lý khiếu nại | 2 |
| PB13 | Hỗ trợ Chatbot AI | Học viên/Gia sư | Hỗ trợ học tập qua AI | Chatbot trả lời câu hỏi, cung cấp tài liệu | 2 |
| PB14 | Quản lý tài khoản | Admin | Quản lý user trên hệ thống | Có thể khóa/mở tài khoản, chỉnh sửa thông tin | 1 |
| PB15 | Quản lý báo cáo và phân tích dữ liệu | Admin | Xem phân tích hệ thống | Thống kê người dùng, giao dịch, doanh thu | 3 |
| PB16 | Quản lý kho tài liệu phản hồi | Admin | Xem phân tích tài liệu lưu trữ phản hồi | Thống kê các phản hồi, đánh giá và tài liệu đã lưu trữ | 3 |

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

### 2.1.1 PB01-Đăng ký tài khoản

|  |  |
| --- | --- |
| **ID** | PB01 |
| **Tiêu đề** | Đăng ký tài khoản |
| **Mô tả** | Người dùng có thể đăng ký tài khoản để truy cập và sử dụng dịch vụ hệ thống. |
| **Các bước thực hiện** | 1. Thiết kế giao diện đăng ký tài khoản trên trang web. 2. Người dùng nhập thông tin cá nhân (họ tên, email, số điện thoại, mật khẩu). 3. Hệ thống kiểm tra tính hợp lệ của thông tin:  * Email chưa tồn tại trong hệ thống. * Số điện thoại hợp lệ. * Mật khẩu đáp ứng tiêu chí bảo mật (ít nhất 8 ký tự, có chữ hoa, chữ thường và số).  1. Nếu hợp lệ, hệ thống lưu thông tin người dùng vào cơ sở dữ liệu. 2. Hệ thống gửi email xác nhận tài khoản. 3. Người dùng truy cập email và nhấp vào liên kết xác nhận. 4. Hệ thống kích hoạt tài khoản và hiển thị thông báo đăng ký thành công. |
| **Điều kiện trước** | Người dùng chưa có tài khoản trong hệ thống. |
| **Điều kiện sau** | Người dùng đăng ký thành công và có thể đăng nhập. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (thiết kế giao diện đăng ký, gọi API) * **Backend:** Node.js (Express.js) (xử lý logic đăng ký, gửi email xác nhận) * **Database:** MongoDB (lưu trữ thông tin người dùng) * **API:** RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình |

### 2.1.2 PB02-Đăng nhập

|  |  |
| --- | --- |
| **ID** | PB02 |
| **Tiêu đề** | Đăng nhập |
| **Mô tả** | Người dùng có thể đăng nhập vào hệ thống để sử dụng các tính năng. |
| **Các bước thực hiện** | |  | | --- | | 1. Thiết kế giao diện trang đăng nhập với các trường nhập email và mật khẩu. | | 1. Xử lý logic kiểm tra thông tin đăng nhập. | | 1. Gửi yêu cầu đăng nhập đến API kiểm tra thông tin. | | 1. Nếu thông tin hợp lệ, hệ thống xác thực và cấp quyền truy cập. | | 1. Nếu thông tin không hợp lệ, hiển thị thông báo lỗi cho người dùng. | |
| **Điều kiện trước** | Người dùng đã có tài khoản hợp lệ trong hệ thống. |
| **Điều kiện sau** | Người dùng đăng nhập thành công và có quyền sử dụng hệ thống. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (thiết kế giao diện đăng nhập, gọi API). * **Backend:** Node.js (xử lý đăng nhập, kiểm tra thông tin người dùng). * **Database:** MongoDB (lưu trữ thông tin tài khoản). * **API:** RESTful API (gửi và nhận dữ liệu từ backend). |
| **Độ phức tạp** | Trung bình |

### 2.1.3 PB05-Tìm kiếm gia sư

|  |  |
| --- | --- |
| **ID** | PB05 |
| **Tiêu đề** | Tìm kiếm gia sư |
| **Mô tả** | Học viên có thể tìm kiếm gia sư phù hợp dựa trên môn học, địa điểm và đánh giá. |
| **Các bước thực hiện** | 1. Thiết kế giao diện trang tìm kiếm gia sư với ô nhập từ khóa và bộ lọc (môn học, địa điểm, đánh giá). 2. Xử lý logic tìm kiếm từ khóa và áp dụng bộ lọc. 3. Gửi yêu cầu tìm kiếm đến API và lấy danh sách gia sư phù hợp. 4. Hiển thị danh sách gia sư theo kết quả tìm kiếm. 5. Học viên có thể chọn một gia sư để xem thông tin chi tiết. |
| **Điều kiện trước** | Học viên đã đăng nhập vào hệ thống. |
| **Điều kiện sau** | Học viên thấy danh sách gia sư phù hợp với yêu cầu. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (thiết kế giao diện tìm kiếm, gọi API). * **Backend:** Node.js (xử lý yêu cầu tìm kiếm, lấy dữ liệu gia sư). * **Database:** MongoDB (lưu trữ danh sách gia sư, đánh giá). * **API:** RESTful API (gửi và nhận dữ liệu từ backend). |
| **Độ phức tạp** | Cao |

### 2.1.4 PB07-Đặt lịch học

|  |  |
| --- | --- |
| **ID** | PB07 |
| **Tiêu đề** | Đặt lịch học |
| **Mô tả** | Học viên có thể đặt lịch học với gia sư theo thời gian phù hợp. |
| **Các bước thực hiện** | 1. Thiết kế giao diện trang đặt lịch với danh sách gia sư và lịch trống. 2. Học viên chọn gia sư và thời gian mong muốn. 3. Gửi yêu cầu đặt lịch đến API để xử lý. 4. Hệ thống gửi thông báo đến gia sư để xác nhận hoặc đề xuất thời gian khác. 5. Nếu gia sư xác nhận, hệ thống cập nhật lịch học và gửi thông báo cho học viên. 6. Nếu gia sư từ chối hoặc đề xuất thời gian khác, học viên có thể điều chỉnh lại lịch đặt. |
| **Điều kiện trước** | Học viên đã đăng nhập và có tài khoản hợp lệ.  1. Gia sư có sẵn lịch dạy để lựa chọn. |
| **Điều kiện sau** | 1. Lịch học được thiết lập thành công. 2. Học viên và gia sư đều nhận được thông báo về lịch học. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (thiết kế giao diện đặt lịch, gọi API). * **Backend:** Node.js (xử lý logic đặt lịch, gửi thông báo). * **Database:** MongoDB (lưu trữ thông tin lịch học, trạng thái xác nhận). * **API:** RESTful API (gửi và nhận dữ liệu đặt lịch từ backend). |
| **Độ phức tạp** | Cao |

### 2.1.5 PB10-Thanh toán online

|  |  |
| --- | --- |
| **ID** | PB10 |
| **Tiêu đề** | Thanh toán online |
| **Mô tả** | Học viên có thể thanh toán online qua nhiều phương thức khác nhau. |
| **Các bước thực hiện** | 1. Thiết kế giao diện trang thanh toán với danh sách các gói học phí. 2. Học viên chọn gói học phí cần thanh toán và phương thức thanh toán (thẻ tín dụng, ví điện tử, chuyển khoản). 3. Hệ thống xử lý giao dịch thông qua cổng thanh toán tích hợp. 4. Hệ thống gửi thông báo về trạng thái giao dịch (thành công/thất bại). 5. Nếu giao dịch thành công, hệ thống cập nhật trạng thái học phí và cho phép học viên tham gia lớp học. 6. Nếu giao dịch thất bại, hệ thống hiển thị thông báo lỗi và cho phép học viên thử lại. |
| **Điều kiện trước** | 1. Học viên đã đăng nhập vào hệ thống. 2. Học viên có lịch học cần thanh toán. 3. Hệ thống đã tích hợp các cổng thanh toán trực tuyến. |
| **Điều kiện sau** | 1. Thanh toán được xác nhận thành công. 2. Học viên có thể tham gia lớp học theo lịch đã đặt. 3. Hệ thống ghi nhận và cập nhật trạng thái thanh toán của học viên. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện thanh toán, hiển thị trạng thái giao dịch). * **Backend:** Node.js (xử lý giao dịch, kết nối với cổng thanh toán). * **Database:** MongoDB (lưu trữ lịch sử thanh toán, trạng thái giao dịch). * **API:** RESTful API (gửi và nhận dữ liệu từ hệ thống thanh toán). * **Cổng thanh toán:** Tích hợp các dịch vụ như PayPal, Momo, VNPAY, Stripe. |
| **Độ phức tạp** | Cao |

### 2.1.6 PB06-Xem chi tiết gia sư

|  |  |
| --- | --- |
| **ID** | PB06 |
| **Tiêu đề** | Xem chi tiết gia sư |
| **Mô tả** | Học viên có thể xem thông tin chi tiết về gia sư trước khi đặt lịch học. |
| **Các bước thực hiện** | 1. Học viên chọn một gia sư từ danh sách kết quả tìm kiếm. 2. Hệ thống hiển thị thông tin chi tiết của gia sư, bao gồm:  * Họ tên, kinh nghiệm giảng dạy, bằng cấp. * Môn học giảng dạy. * Đánh giá và phản hồi từ học viên khác.  1. Học viên có thể chọn nút "Đặt lịch" để tiếp tục đặt lịch học với gia sư. |
| **Điều kiện trước** | 1. Học viên đã tìm kiếm gia sư trong hệ thống. 2. Hệ thống có dữ liệu về gia sư và đánh giá từ học viên khác. |
| **Điều kiện sau** | Học viên có thể quyết định đặt lịch học với gia sư hoặc tiếp tục tìm kiếm gia sư khác. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện hiển thị thông tin gia sư, nút đặt lịch). * **Backend:** Node.js (truy xuất thông tin gia sư từ cơ sở dữ liệu). * **Database:** MongoDB (lưu trữ thông tin gia sư, đánh giá từ học viên). * **API:** RESTful API (gửi yêu cầu lấy thông tin gia sư từ backend). |
| **Độ phức tạp** | Trung bình |

### PB08 – Tra cứu lịch học

|  |  |
| --- | --- |
| ID | PB08 |
| **Tiêu đề** | Tra cứu lịch học |
| **Mô tả** | Người dùng có thể xem danh sách các buổi học đã đặt, bao gồm thông tin về ngày, giờ, gia sư và trạng thái buổi học. |
| **Các bước thực hiện** | 1. Người dùng truy cập trang **"Lịch học"**. 2. Hệ thống hiển thị danh sách các buổi học với thông tin:  * Ngày, giờ học. * Gia sư giảng dạy. * Môn học. * Trạng thái buổi học (đã xác nhận, đang chờ, đã hủy).  1. Người dùng có thể **lọc, tìm kiếm lịch học** theo ngày, môn học hoặc gia sư. 2. Nhấn vào một buổi học cụ thể để xem chi tiết. |
| **Điều kiện trước** | 1. Người dùng đã đăng nhập vào hệ thống. 2. Người dùng đã có ít nhất một buổi học được đặt trước. |
| **Điều kiện sau** | Người dùng có thể xem thông tin chi tiết của lịch học. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (hiển thị giao diện tra cứu lịch học). * **Backend:** Node.js (xử lý yêu cầu lấy danh sách lịch học). * **Database:** MongoDB (lưu trữ thông tin lịch học). * **API:** RESTful API (gửi yêu cầu lấy dữ liệu lịch học). |
| **Độ phức tạp** | Trung bình |

### 2.1.8 PB04-Quản lý thông tin cá nhân

|  |  |
| --- | --- |
| ID | PB04 |
| **Tiêu đề** | Quản lý thông tin cá nhân |
| **Mô tả** | Người dùng có thể chỉnh sửa hồ sơ cá nhân để cập nhật thông tin mới nhất. |
| **Các bước thực hiện** | 1. Người dùng truy cập trang "Thông tin cá nhân". 2. Có thể chỉnh sửa các thông tin sau:  * Họ và tên. * Email (phải đúng định dạng). * Ảnh đại diện. * Số điện thoại.  1. Nhấn "Lưu thay đổi", hệ thống kiểm tra tính hợp lệ của dữ liệu. 2. Nếu hợp lệ, hệ thống cập nhật thông tin thành công và hiển thị thông báo. |
| **Điều kiện trước** | Người dùng đã đăng nhập vào hệ thống. |
| **Điều kiện sau** | Thông tin cá nhân được cập nhật thành công trên hệ thống. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện chỉnh sửa thông tin cá nhân). * **Backend:** Node.js (xử lý logic cập nhật thông tin). * **Database:** MongoDB (lưu trữ thông tin cá nhân). * **API:** RESTful API (gửi yêu cầu cập nhật dữ liệu lên backend). |
| **Độ phức tạp** | Trung bình |

### 2.1.9 PB09-Quản lý lịch dạy (Gia sư)

|  |  |
| --- | --- |
| ID | PB09 |
| **Tiêu đề** | Quản lý lịch dạy |
| **Mô tả** | Gia sư có thể xem và quản lý lịch dạy của mình trên hệ thống. |
| **Các bước thực hiện** | 1. Người dùng truy cập trang "Lịch dạy". 2. Hệ thống hiển thị danh sách lịch dạy sắp tới. 3. Gia sư có thể xác nhận hoặc từ chối yêu cầu thay đổi. 4. Nếu thay đổi được chấp nhận, hệ thống cập nhật lịch dạy mới. |
| **Điều kiện trước** | Người dùng đã đăng nhập vào hệ thống. |
| **Điều kiện sau** | Lịch dạy được cập nhật theo yêu cầu thay đổi. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (hiển thị và thao tác lịch dạy). * **Backend:** Node.js (xử lý logic quản lý lịch dạy). * **Database:** MongoDB (lưu trữ thông tin lịch dạy). * **API:** RESTful API (gửi và nhận yêu cầu cập nhật lịch dạy). |
| **Độ phức tạp** | Trung bình |

### 2.1.10 PB13-Hỗ trợ Chatbot AI

|  |  |
| --- | --- |
| **ID** | PB12 |
| **Tiêu đề** | Hỗ trợ Chatbot AI |
| **Mô tả** | Chatbot hỗ trợ học viên và gia sư bằng cách cung cấp tài liệu và trả lời câu hỏi. |
| **Các bước thực hiện** | 1. Người dùng truy cập chatbot trên trang chủ. 2. Nhập câu hỏi hoặc từ khóa liên quan đến môn học. 3. Chatbot phản hồi với câu trả lời hoặc gợi ý tài liệu phù hợp. 4. Nếu chatbot không có câu trả lời, hệ thống có thể chuyển tiếp đến hỗ trợ viên. |
| **Điều kiện trước** | Chatbot đã được tích hợp vào hệ thống. |
| **Điều kiện sau** | Người dùng nhận được thông tin hỗ trợ phù hợp. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (hiển thị giao diện chatbot). * **Backend:** Node.js (xử lý logic chatbot và phản hồi). * **Database:** MongoDB (lưu trữ dữ liệu câu hỏi và phản hồi). * **AI:** (để xử lý câu hỏi). * **API:** RESTful API (kết nối chatbot với hệ thống). |
| **Độ phức tạp** | Cao |

### 2.1.11 PB03-Quên mật khẩu

|  |  |
| --- | --- |
| **ID** | PB03 |
| **Tiêu đề** | Quên mật khẩu |
| **Mô tả** | Người dùng có thể khôi phục mật khẩu nếu quên để tiếp tục sử dụng hệ thống. |
| **Các bước thực hiện** | 1. Người dùng truy cập trang "Quên mật khẩu". 2. Nhập email đăng ký, hệ thống kiểm tra tính hợp lệ của email. 3. Nếu hợp lệ, hệ thống gửi email xác nhận khôi phục mật khẩu. 4. Người dùng nhấn vào liên kết xác nhận trong email. 5. Nhập mật khẩu mới và xác nhận lại. 6. Nếu hợp lệ, hệ thống cập nhật mật khẩu mới và thông báo thành công. |
| **Điều kiện trước** | 1. Người dùng đã có tài khoản nhưng quên mật khẩu. 2. Email nhập vào phải trùng với email đã đăng ký trước đó. |
| **Điều kiện sau** | 1. Mật khẩu mới được thiết lập thành công. 2. Người dùng có thể đăng nhập lại bằng mật khẩu mới. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (giao diện quên mật khẩu, nhập email, đổi mật khẩu). * **Backend:** Node.js (xử lý logic quên mật khẩu, xác thực email). * **Database:** MongoDB (cập nhật mật khẩu mới của người dùng). * **API:** RESTful API (xử lý yêu cầu đổi mật khẩu). * **Email Service:** Nodemailer (gửi email xác nhận đổi mật khẩu). |
| **Độ phức tạp** | Trung bình |

### 2.1.12 PB11-Đánh giá gia sư

|  |  |
| --- | --- |
| **ID** | PB10 |
| **Tiêu đề** | Đánh giá gia sư |
| **Mô tả** | Học viên có thể đánh giá chất lượng của gia sư sau mỗi buổi học để giúp các học viên khác lựa chọn phù hợp hơn. |
| **Các bước thực hiện** | 1. Học viên truy cập trang đánh giá gia sư. 2. Chọn gia sư đã có buổi học. 3. Nhập đánh giá bằng số sao và bình luận. 4. Hệ thống lưu đánh giá và cập nhật điểm trung bình. |
| **Điều kiện trước** | Học viên đã tham gia ít nhất một buổi học với gia sư. |
| **Điều kiện sau** | 1. Đánh giá được lưu và hiển thị trên hồ sơ gia sư. 2. Điểm đánh giá trung bình của gia sư được cập nhật. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (hiển thị giao diện đánh giá, nhập số sao và bình luận). * **Backend:** Node.js (xử lý lưu đánh giá, tính toán điểm trung bình). * **Database:** MongoDB (lưu trữ đánh giá, liên kết với hồ sơ gia sư). * **API:** RESTful API (gửi và nhận dữ liệu đánh giá). |
| **Độ phức tạp** | Trung bình |

### 2.1.13 PB12-Quản lý thanh toán (Admin)

|  |  |
| --- | --- |
| **ID** | PB11 |
| **Tiêu đề** | Quản lý thanh toán |
| **Mô tả** | Admin có thể kiểm soát các giao dịch thanh toán trên hệ thống để đảm bảo tính minh bạch và hỗ trợ xử lý khiếu nại khi cần. |
| **Các bước thực hiện** | 1. Admin truy cập trang quản lý thanh toán. 2. Hệ thống hiển thị danh sách các giao dịch. 3. Admin có thể lọc giao dịch theo trạng thái (thành công, thất bại, hoàn tiền). 4. Admin có thể xem chi tiết giao dịch và xử lý khiếu nại hoặc hoàn tiền khi cần. |
| **Điều kiện trước** | Hệ thống có dữ liệu thanh toán từ các giao dịch trước đó. |
| **Điều kiện sau** | 1. Giao dịch được kiểm soát hợp lệ. 2. Nếu có khiếu nại, hệ thống ghi nhận và xử lý theo quy trình. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (hiển thị danh sách giao dịch, bộ lọc, chi tiết thanh toán). * **Backend:** Node.js (xử lý dữ liệu thanh toán, cập nhật trạng thái giao dịch). * **Database:** MongoDB (lưu trữ lịch sử thanh toán, trạng thái giao dịch). * **API:** RESTful API (gửi và nhận dữ liệu thanh toán từ backend). |
| **Độ phức tạp** | Cao |



### 2.1.14 PB14-Quản lý tài khoản (Admin)

|  |  |
| --- | --- |
| **ID** | PB13 |
| **Tiêu đề** | Quản lý tài khoản |
| **Mô tả** | Admin có thể quản lý các tài khoản người dùng, bao gồm xem danh sách, khóa/mở tài khoản và chỉnh sửa thông tin khi cần. |
| **Các bước thực hiện** | 1. Admin truy cập trang quản lý tài khoản. 2. Hệ thống hiển thị danh sách tài khoản người dùng. 3. Admin có thể tìm kiếm tài khoản theo tên, email hoặc vai trò. 4. Admin có thể thực hiện các thao tác sau:  * Khóa/mở tài khoản. * Chỉnh sửa thông tin tài khoản (tên, email, quyền truy cập).  1. Hệ thống lưu lại thay đổi và cập nhật trạng thái tài khoản. |
| **Điều kiện trước** | 1. Hệ thống có danh sách tài khoản người dùng. 2. Admin có quyền quản lý tài khoản. |
| **Điều kiện sau** | 1. Tài khoản được quản lý hợp lệ. 2. Nếu tài khoản bị khóa, người dùng không thể đăng nhập vào hệ thống. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (hiển thị danh sách tài khoản, thao tác quản lý). * **Backend:** Node.js (xử lý quản lý tài khoản, cập nhật trạng thái). * **Database:** MongoDB (lưu trữ thông tin tài khoản, trạng thái tài khoản). * **API:** RESTful API (gửi và nhận dữ liệu từ backend). |
| **Độ phức tạp** | Trung bình |

### 2.1.15 PB15-Quản lý báo cáo và phân tích dữ liệu

|  |  |
| --- | --- |
| **ID** | PB14 |
| **Tiêu đề** | Quản lý báo cáo và phân tích dữ liệu |
| **Mô tả** | Admin có thể xem các báo cáo thống kê về hệ thống, bao gồm số lượng người dùng, giao dịch, doanh thu, và có thể lọc hoặc xuất báo cáo khi cần. |
| **Các bước thực hiện** | 1. Admin truy cập trang báo cáo. 2. Hệ thống hiển thị bảng thống kê số lượng người dùng, giao dịch và doanh thu. 3. Admin có thể lọc báo cáo theo khoảng thời gian hoặc danh mục cụ thể. 4. Hệ thống hiển thị dữ liệu báo cáo dưới dạng bảng và biểu đồ trực quan. 5. Admin có thể chọn xuất báo cáo dưới dạng **Excel** hoặc **PDF**. |
| **Điều kiện trước** | Hệ thống đã thu thập đủ dữ liệu từ người dùng và giao dịch. |
| **Điều kiện sau** | 1. Báo cáo được hiển thị hoặc xuất thành công. 2. Admin có thể tải báo cáo về dưới dạng file Excel/PDF. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (hiển thị dữ liệu báo cáo, biểu đồ, chức năng lọc). * **Backend:** Node.js (xử lý truy vấn dữ liệu thống kê, xuất file báo cáo). * **Database:** MongoDB (lưu trữ dữ liệu giao dịch, người dùng). * **API:** RESTful API (lấy dữ liệu báo cáo từ backend). * **Thư viện xuất báo cáo:** ExcelJS (xuất file Excel), jsPDF (xuất file PDF). |
| **Độ phức tạp** | Cao |

### PB16 – Quản lý kho tài liệu phản hồi

|  |  |
| --- | --- |
| **ID** | PB16 |
| **Tiêu đề** | Quản lý kho tài liệu phản hồi |
| **Mô tả** | Admin có thể quản lý kho tài liệu phản hồi của hệ thống, bao gồm phản hồi từ người dùng, đánh giá về gia sư và các tài liệu hỗ trợ học tập. Hệ thống cho phép lọc, tìm kiếm và xuất báo cáo khi cần. |
| **Các bước thực hiện** | 1. **Admin** truy cập trang **"Quản lý kho tài liệu phản hồi"**. 2. Hệ thống hiển thị danh sách các phản hồi và tài liệu lưu trữ, bao gồm:  * Tên người gửi phản hồi. * Nội dung phản hồi. * Ngày gửi. * Loại phản hồi (đánh giá gia sư, khiếu nại, góp ý, hỗ trợ học tập).  1. Admin có thể **lọc phản hồi** theo thời gian, loại phản hồi hoặc người gửi. 2. Admin có thể xem chi tiết từng phản hồi và phản hồi lại nếu cần. 3. Hệ thống cung cấp **tùy chọn xuất báo cáo** về phản hồi dưới dạng Excel hoặc PDF. |
| **Điều kiện trước** | Hệ thống đã thu thập đủ dữ liệu từ người dùng. |
| **Điều kiện sau** | 1. Admin có thể xem, lọc hoặc tìm kiếm các phản hồi một cách dễ dàng. 2. Admin có thể xuất báo cáo phản hồi thành công dưới dạng file Excel/PDF. |
| **Công nghệ sử dụng** | * **Frontend:** React.js (hiển thị danh sách phản hồi, giao diện quản lý). * **Backend:** Node.js (xử lý truy vấn dữ liệu phản hồi, xuất file báo cáo). * **Database:** MongoDB (lưu trữ thông tin phản hồi). * **API:** RESTful API (gửi yêu cầu lấy dữ liệu phản hồi từ backend). * **Thư viện xuất báo cáo:** ExcelJS (xuất file Excel), jsPDF (xuất file PDF). |
| **Độ phức tạp** | Cao |

### Phân tích về Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint | Function | Started Date | Due Date |
| Sprint 1 | Develop | 08/04/2025 | 05/05/2025 |
| Sprint 2 | Develop | 05/05/2025 | 30/05/2025 |

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. |