

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



## LẬP TRÌNH WEB (CO3049)

---

Bài tập lớn

### THIẾT KẾ GIAO DIỆN VÀ XÂY DỰNG CÁC TÍNH NĂNG CƠ BẢN CHO WEBSITE ĐẶT PHÒNG HỌP PROMEET

---

Giảng viên hướng dẫn: NGUYỄN HỮU HIẾU

Lớp: L02

Sinh viên: Trần Quang Tác - 2212962

Email liên hệ: tac.tranquang@hcmut.edu.vn



## Mục lục

<b>1 Giới thiệu</b>	<b>3</b>
<b>2 Cơ sở lý thuyết</b>	<b>3</b>
2.1 HTML, CSS, JavaScript . . . . .	3
2.2 PHP . . . . .	4
2.3 MySQL . . . . .	4
2.4 Bootstrap 5 . . . . .	5
2.5 Mazer Template . . . . .	5
2.6 SweetAlert2 . . . . .	6
2.7 AOS (Animate On Scroll) . . . . .	6
2.8 jQuery . . . . .	7
2.9 Fetch API . . . . .	7
2.10 TinyMCE . . . . .	8
2.11 Chart.js . . . . .	8
2.12 Simple DataTables . . . . .	9
2.13 Session và Cookie trong PHP . . . . .	9
2.14 Các lỗ hổng bảo mật phổ biến . . . . .	10
2.15 SEO (Search Engine Optimization) . . . . .	10
<b>3 Thiết kế ứng dụng</b>	<b>10</b>
3.1 Thiết kế cơ sở dữ liệu . . . . .	10
3.2 Mô tả chi tiết các thực thể . . . . .	11
3.2.1 Thực thể users . . . . .	11
3.2.2 Thực thể rooms . . . . .	12
3.2.3 Thực thể bookings . . . . .	13
3.2.4 Thực thể booking_slots . . . . .	13
3.2.5 Thực thể booking_status_history . . . . .	13
3.2.6 Thực thể reviews . . . . .	14
3.2.7 Thực thể images . . . . .	14
3.2.8 Thực thể remember_tokens . . . . .	15
3.2.9 Thực thể room_stats . . . . .	15
3.3 Sơ đồ Usecase Diagram . . . . .	15
3.4 Đặc tả Usecase . . . . .	17
3.4.1 <User> Đặt phòng họp (UC_01) . . . . .	17
3.4.2 <User> Chính sửa thông tin cá nhân (UC_02) . . . . .	18
3.4.3 <Admin> Thêm phòng họp mới (UC_01) . . . . .	19
3.4.4 <Admin> Cập nhật trạng thái lịch đặt (UC_02) . . . . .	20
3.5 Activity Diagram . . . . .	21
3.5.1 Đặt phòng họp (User) . . . . .	21
3.5.2 Duyệt / Quản lý đơn đặt phòng (Admin) . . . . .	22
3.6 Kiến trúc hệ thống . . . . .	23
3.7 Design Pattern . . . . .	25



---

<b>4</b>	<b>Hiện thực</b>	<b>28</b>
4.1	Về phía khách (chưa đăng nhập) . . . . .	28
4.2	Về phía người dùng (đã đăng nhập) . . . . .	36
4.3	Về phía quản trị viên . . . . .	42
<b>5</b>	<b>Cách cài đặt ứng dụng</b>	<b>51</b>
<b>6</b>	<b>Nhiệm vụ của từng thành viên</b>	<b>54</b>
<b>7</b>	<b>Source code BTL</b>	<b>55</b>



## 1 Giới thiệu

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ tại Việt Nam, website đã trở thành một thành phần thiết yếu trong hoạt động của các doanh nghiệp. Website không chỉ là kênh thông tin chính thức để giới thiệu về doanh nghiệp, sản phẩm và dịch vụ mà còn là nền tảng để tương tác với khách hàng, đối tác và cộng đồng một cách nhanh chóng, thuận tiện và chuyên nghiệp.

Đặc biệt tại các đô thị lớn như Thành phố Hồ Chí Minh – nơi nhu cầu về không gian làm việc chung và phòng họp theo giờ đang ngày càng tăng – các doanh nghiệp cung cấp dịch vụ coworking space và cho thuê phòng họp cần có nền tảng trực tuyến chuyên nghiệp để quản lý đặt chỗ, hiển thị thông tin dịch vụ, tiếp nhận phản hồi và hỗ trợ thanh toán.

Đề tài xây dựng website **ProMeet** – một nền tảng đặt phòng họp trực tuyến – nhằm mô phỏng cách một doanh nghiệp vận hành dịch vụ cho thuê không gian làm việc thông qua hệ thống web. Website bao gồm các chức năng chính như: giới thiệu dịch vụ, danh sách và chi tiết phòng họp, bộ lọc theo nhu cầu, đặt phòng theo thời gian thực, đánh giá người dùng, quản lý đơn hàng và hệ thống quản trị dành cho admin.

Qua việc phát triển ProMeet, sinh viên có cơ hội áp dụng kiến thức lập trình web vào mô hình doanh nghiệp thực tế, đồng thời tìm hiểu sâu hơn về kiến trúc website, bảo mật ứng dụng web, tối ưu SEO và trải nghiệm người dùng trong môi trường số hoá hiện đại.

## 2 Cơ sở lý thuyết

### 2.1 HTML, CSS, JavaScript

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu tiêu chuẩn dùng để xây dựng cấu trúc nội dung cho các trang web. HTML định nghĩa các phần tử như tiêu đề, đoạn văn, liên kết, hình ảnh, biểu mẫu và các thành phần giao diện khác, đóng vai trò là "bộ xương" của một website.

CSS (Cascading Style Sheets) là ngôn ngữ định kiểu cho các phần tử HTML, cho phép lập trình viên thiết kế bố cục, màu sắc, phông chữ, hiệu ứng chuyển động và các yếu tố thị giác khác. CSS hỗ trợ các kỹ thuật hiện đại như responsive design để tối ưu hiển thị trên nhiều loại thiết bị.

JavaScript là ngôn ngữ lập trình phía client giúp bổ sung tính năng tương tác cho trang web. Nó cho phép xử lý sự kiện, thay đổi nội dung động, thực hiện kiểm tra dữ liệu phía trình duyệt, và gọi API bất đồng bộ (AJAX hoặc Fetch API). JavaScript có thể hoạt động độc lập hoặc kết hợp với các thư viện như jQuery để đơn giản hóa thao tác DOM và xử lý sự kiện.

#### Ưu điểm:

- HTML, CSS, JavaScript là các công nghệ cốt lõi của web, được chuẩn hóa và hỗ trợ rộng rãi trên mọi trình duyệt.
- Có cộng đồng phát triển lớn, nhiều tài liệu học tập và thư viện hỗ trợ.
- Dễ tiếp cận cho người mới bắt đầu và có thể mở rộng với các framework như React, Vue, Angular,...

#### Nhược điểm:

- HTML là ngôn ngữ tĩnh, không có khả năng xử lý logic hoặc tương tác mà không kết hợp với JavaScript.
- CSS thường có thể khó kiểm soát khi dự án lớn nếu không có cấu trúc rõ ràng hoặc sử dụng framework hỗ trợ.
- JavaScript có thể bị vô hiệu hóa phía người dùng, và mã JavaScript không được kiểm soát chặt có thể gây ra rủi ro bảo mật như XSS.



## 2.2 PHP

PHP (viết tắt của "PHP: Hypertext Preprocessor") là một ngôn ngữ lập trình kịch bản phía máy chủ (server-side scripting language), được thiết kế chủ yếu để phát triển các ứng dụng web động. PHP được nhúng trực tiếp vào mã HTML, và được máy chủ web xử lý để tạo ra nội dung trang web động.

### Ưu điểm:

- Miễn phí, mã nguồn mở, và có cộng đồng hỗ trợ lớn.
- Cú pháp đơn giản, dễ học, dễ tiếp cận đối với người mới bắt đầu.
- Tích hợp dễ dàng với nhiều hệ quản trị cơ sở dữ liệu như MySQL, PostgreSQL,...
- Hỗ trợ sẵn cho các chức năng web như xử lý biểu mẫu, session, cookie, tải tệp, gửi email,...

### Nhược điểm:

- Cấu trúc mã dễ trở nên lộn xộn nếu không tổ chức tốt hoặc không áp dụng các mô hình chuẩn (như MVC).
- Một số tính năng hướng đối tượng còn hạn chế so với các ngôn ngữ hiện đại như Java, C#.
- Dễ bị khai thác các lỗ hổng bảo mật (như SQL Injection, XSS, CSRF) nếu không xử lý cẩn thận dữ liệu đầu vào.

PHP vẫn là một trong những ngôn ngữ phổ biến nhất để xây dựng các hệ thống web vừa và nhỏ nhờ vào tính linh hoạt, hiệu năng tương đối và khả năng triển khai rộng rãi trên các môi trường máy chủ.

## 2.3 MySQL

MySQL là một hệ quản trị cơ sở dữ liệu (DBMS) mã nguồn mở, sử dụng mô hình cơ sở dữ liệu quan hệ (RDBMS). MySQL hỗ trợ lưu trữ dữ liệu theo dạng bảng và cung cấp các công cụ mạnh mẽ để thực hiện các thao tác như truy vấn, cập nhật, xóa và quản lý dữ liệu. Ngôn ngữ truy vấn được sử dụng chủ yếu trong MySQL là SQL (Structured Query Language).

MySQL là một trong những cơ sở dữ liệu phổ biến nhất hiện nay, được sử dụng rộng rãi trong các ứng dụng web nhờ vào khả năng tương thích với nhiều ngôn ngữ lập trình, tính ổn định và khả năng xử lý khối lượng lớn dữ liệu với hiệu suất cao. MySQL có thể được sử dụng trong cả các ứng dụng quy mô nhỏ đến các hệ thống phức tạp với yêu cầu cao về độ tin cậy.

### Ưu điểm:

- MySQL là phần mềm mã nguồn mở, miễn phí và có cộng đồng phát triển lớn.
- Hỗ trợ tốc độ xử lý cao với khả năng tối ưu các truy vấn phức tạp và quy mô lớn.
- Có khả năng quản lý dữ liệu quan hệ tốt và dễ dàng duy trì tính toàn vẹn của dữ liệu thông qua các cơ chế như khóa ngoại (foreign key), chỉ mục (index) và giao dịch (transaction).
- Tích hợp dễ dàng với các ngôn ngữ lập trình phổ biến như PHP, Python, Java, Ruby, Node.js.
- Có nhiều công cụ quản lý như phpMyAdmin, MySQL Workbench giúp quản lý và tối ưu cơ sở dữ liệu dễ dàng.

### Nhược điểm:



- MySQL có thể không tối ưu cho các ứng dụng yêu cầu xử lý dữ liệu phi quan hệ (NoSQL), do vậy không phải là lựa chọn lý tưởng cho tất cả các loại dữ liệu.
- Một số tính năng nâng cao của MySQL (như các hàm window hoặc CTE) không hoàn toàn mạnh mẽ như các hệ quản trị cơ sở dữ liệu khác như PostgreSQL.
- MySQL không có hỗ trợ sẵn cho tính năng phân tán dữ liệu, khiến cho việc mở rộng theo chiều ngang (horizontal scaling) trở nên phức tạp hơn.
- Một số tính năng của MySQL chỉ có sẵn trong phiên bản trả phí (ví dụ, MySQL Enterprise Edition).

## 2.4 Bootstrap 5

Bootstrap 5 là một framework CSS mã nguồn mở được phát triển bởi Twitter, giúp xây dựng giao diện web nhanh chóng và dễ dàng. Nó cung cấp một bộ các thành phần giao diện người dùng (UI components), như nút, thanh điều hướng, form, bảng, v.v., giúp tiết kiệm thời gian phát triển và duy trì tính nhất quán trong giao diện người dùng. Một trong những tính năng mạnh mẽ của Bootstrap là khả năng tạo giao diện responsive – tự động điều chỉnh giao diện của website sao cho phù hợp với mọi kích thước màn hình, từ điện thoại di động đến máy tính để bàn.

### Ưu điểm:

- Tăng tốc quá trình phát triển giao diện web.
- Hỗ trợ đa nền tảng, đảm bảo giao diện hiển thị tốt trên mọi thiết bị.
- Cộng đồng phát triển lớn, dễ dàng tìm kiếm tài liệu và hỗ trợ.

### Nhược điểm:

- Giao diện có thể trông giống nhau trên nhiều trang web, nếu không tùy chỉnh kỹ lưỡng.
- Khối lượng mã CSS và JavaScript của Bootstrap có thể làm tăng kích thước trang web.

## 2.5 Mazer Template

Mazer Template là một template giao diện quản trị (Admin UI) hiện đại được xây dựng bằng HTML, CSS, và JavaScript, giúp tạo ra các trang quản trị dễ sử dụng và thân thiện với người dùng. Mazer cung cấp một bộ giao diện có sẵn cho các ứng dụng quản trị, bao gồm các phần như bảng điều khiển, biểu đồ, danh sách, và các thành phần giao diện khác. Template này rất thích hợp cho các ứng dụng cần quản lý dữ liệu hoặc các hệ thống backend.

### Ưu điểm:

- Giao diện trực quan, dễ sử dụng cho các ứng dụng quản trị.
- Thiết kế đẹp mắt và dễ tùy chỉnh.
- Tích hợp nhiều tính năng hỗ trợ cho việc quản lý dữ liệu và người dùng.

### Nhược điểm:

- Template có thể yêu cầu điều chỉnh để phù hợp với các yêu cầu cụ thể của hệ thống.
- Một số tính năng có thể dư thừa hoặc không cần thiết đối với từng dự án.



## 2.6 SweetAlert2

**SweetAlert2** là một thư viện JavaScript giúp tạo các hộp thoại thông báo (popup) đẹp mắt và dễ sử dụng. Thư viện này hỗ trợ các kiểu thông báo như cảnh báo, thành công, lỗi, thông tin, v.v., và có thể tùy chỉnh cao. SweetAlert2 sử dụng giao diện hiện đại và có khả năng tạo ra các thông báo động với nhiều hiệu ứng đẹp mắt.

### Thư viện sử dụng:

- **sweetalert2@11:** Phiên bản mới nhất của SweetAlert2, hỗ trợ đầy đủ các tính năng, bao gồm tùy chỉnh giao diện, hiệu ứng động, v.v.

### Ưu điểm:

- Hộp thoại thông báo đẹp và dễ sử dụng.
- Cung cấp nhiều tùy chọn, bao gồm các hiệu ứng động và tính năng tương tác.
- Hỗ trợ dễ dàng tùy chỉnh theo nhu cầu của ứng dụng.

### Nhược điểm:

- Thư viện khá nặng và có thể làm tăng kích thước trang web nếu sử dụng quá nhiều hộp thoại.
- Các hiệu ứng động có thể làm giảm hiệu suất khi sử dụng trên các thiết bị yếu hoặc trên các trình duyệt cũ.
- Không phải lúc nào cũng cần thiết cho các ứng dụng đơn giản, có thể gây “overkill”.

## 2.7 AOS (Animate On Scroll)

**AOS** là một thư viện JavaScript giúp tạo các hiệu ứng động khi người dùng cuộn trang. Với AOS, các phần tử trên trang sẽ có thể xuất hiện hoặc thay đổi kiểu dáng khi người dùng cuộn đến chúng, giúp giao diện trở nên sinh động hơn.

### Thư viện sử dụng:

- **aos@2.3.1:** Phiên bản này cung cấp các hiệu ứng đơn giản và dễ tùy chỉnh, giúp các phần tử trong trang web có thể chuyển động hoặc thay đổi khi cuộn.

### Ưu điểm:

- Giúp giao diện trang web trở nên sinh động hơn.
- Cung cấp nhiều kiểu hiệu ứng động có sẵn và dễ dàng cấu hình.
- Tăng tính tương tác và hấp dẫn cho người dùng.

### Nhược điểm:

- Có thể làm chậm hiệu suất nếu sử dụng quá nhiều hiệu ứng động hoặc trên các trang có nhiều nội dung.
- Không hỗ trợ tốt trên tất cả các trình duyệt hoặc thiết bị cũ.
- Dễ gây mất tập trung nếu sử dụng không hợp lý hoặc quá mức.



## 2.8 jQuery

**jQuery** là một thư viện JavaScript phổ biến, giúp thao tác với DOM (Document Object Model), xử lý sự kiện, hiệu ứng, và gửi yêu cầu Ajax dễ dàng hơn. Thư viện này từng là tiêu chuẩn de facto cho các ứng dụng web tương tác, đặc biệt trước khi JavaScript thuần trở nên mạnh mẽ hơn.

**Thư viện sử dụng:**

- `jquery-3.6.0.min.js`: Phiên bản jQuery ổn định, giúp thao tác với DOM, sự kiện, và hỗ trợ gửi các yêu cầu bất đồng bộ (Ajax).

**Ưu điểm:**

- Cú pháp đơn giản, giúp viết mã JavaScript dễ hơn, đặc biệt là các thao tác DOM và sự kiện.
- Hỗ trợ tốt trên nhiều trình duyệt, kể cả các trình duyệt cũ.
- Hỗ trợ gửi yêu cầu Ajax thông qua hàm `$.ajax()`, thuận tiện trong việc tương tác với API hoặc server mà không cần tải lại trang.

**Nhược điểm:**

- Kích thước tương đối lớn so với nhu cầu hiện tại nếu chỉ sử dụng một vài chức năng.
- JavaScript hiện đại đã cung cấp các tính năng tương đương (như `querySelector`, `fetch`), khiến jQuery trở nên không cần thiết trong nhiều dự án mới.
- Không còn được cập nhật hoặc phát triển mạnh như trước, dẫn đến nguy cơ bảo trì thấp trong tương lai.

## 2.9 Fetch API

**Fetch API** là một giao diện JavaScript hiện đại dùng để gửi các yêu cầu HTTP bất đồng bộ (asynchronous) từ phía client. Nó thay thế cho XMLHttpRequest (và cả `$.ajax()` của jQuery), với cú pháp đơn giản, rõ ràng, sử dụng Promise.

**Cách sử dụng trong dự án:**

- Dùng để gửi và nhận dữ liệu JSON giữa client và server mà không cần tải lại trang.
- Tương tác với các API RESTful do backend cung cấp.

**Ưu điểm:**

- Giao diện hiện đại, cú pháp rõ ràng, dễ kết hợp với `async/await`.
- Không cần thêm thư viện bên ngoài, giúp giảm kích thước trang web.
- Hỗ trợ Promise, giúp xử lý bất đồng bộ dễ hơn và code gọn gàng hơn.

**Nhược điểm:**

- Không được hỗ trợ trên các trình duyệt quá cũ (như Internet Explorer).
- Phải tự xử lý lỗi HTTP (vì `fetch` không tự ném lỗi với mã phản hồi như 404 hay 500).
- Ít tính năng tiện lợi hơn so với `$.ajax()`, như tự động timeout, retry,...



## 2.10 TinyMCE

**TinyMCE** là một trình soạn thảo WYSIWYG (What You See Is What You Get) cho phép người dùng chỉnh sửa văn bản trực quan, giống như khi soạn thảo văn bản trong các công cụ như Microsoft Word. TinyMCE có thể được tích hợp dễ dàng vào các ứng dụng web và cho phép người dùng tạo ra các nội dung văn bản phong phú với các tính năng như chèn hình ảnh, liên kết, bảng, v.v.

### Thư viện sử dụng:

- **tinymce.min.js:** Trình soạn thảo văn bản WYSIWYG với nhiều tính năng mạnh mẽ, dễ dàng tích hợp vào các ứng dụng web.

### Ưu điểm:

- Tích hợp dễ dàng và linh hoạt vào ứng dụng.
- Cung cấp nhiều tính năng soạn thảo văn bản nâng cao.
- Hỗ trợ đầy đủ các tính năng như chèn hình ảnh, liên kết, bảng, và định dạng văn bản.

### Nhược điểm:

- Kích thước thư viện khá lớn và có thể làm nặng trang web.
- Các tính năng có thể quá phức tạp đối với các yêu cầu đơn giản.
- Nếu cấu hình không cẩn thận, TinyMCE có thể gây lỗi bảo mật liên quan đến việc chèn nội dung từ người dùng.

## 2.11 Chart.js

**Chart.js** là một thư viện JavaScript giúp tạo các biểu đồ đẹp mắt và dễ sử dụng. Thư viện này hỗ trợ nhiều loại biểu đồ như biểu đồ cột, đường, tròn, radar, v.v. và dễ dàng tích hợp vào các ứng dụng web để hiển thị dữ liệu dưới dạng đồ họa.

### Thư viện sử dụng:

- **chart.umd.js:** Phiên bản đầy đủ của Chart.js, cho phép tạo và hiển thị các biểu đồ dữ liệu dễ dàng.

### Ưu điểm:

- Tạo biểu đồ dễ dàng với nhiều loại khác nhau.
- Cung cấp các tùy chọn cấu hình phong phú để tùy chỉnh giao diện biểu đồ.
- Giao diện biểu đồ dễ nhìn và trực quan.

### Nhược điểm:

- Không hỗ trợ tốt cho dữ liệu lớn hoặc biểu đồ phức tạp.
- Có thể gặp khó khăn trong việc tối ưu hóa hiệu suất khi dữ liệu quá lớn.
- Một số tính năng nâng cao yêu cầu phải sử dụng các plugin bổ sung hoặc cấu hình phức tạp.



## 2.12 Simple DataTables

**Simple DataTables** là một thư viện JavaScript nhẹ giúp tạo bảng dữ liệu với tính năng tìm kiếm, phân trang và sắp xếp. Thư viện này dễ dàng tích hợp và hỗ trợ các bảng dữ liệu động, đặc biệt hữu ích khi làm việc với dữ liệu lớn.

### Thư viện sử dụng:

- **simple-datatables.js**: Thư viện giúp tạo bảng dữ liệu với các tính năng cơ bản như tìm kiếm, phân trang và sắp xếp.

### Ưu điểm:

- Tạo bảng dữ liệu với các tính năng cơ bản dễ dàng và nhanh chóng.
- Thư viện nhẹ và dễ dàng tích hợp.
- Hỗ trợ tốt với các bảng dữ liệu có số lượng lớn.

### Nhược điểm:

- Không hỗ trợ các tính năng phức tạp như đồ họa động, thống kê nâng cao.
- Thư viện khá đơn giản, không phù hợp với các ứng dụng yêu cầu giao diện phức tạp hơn.

## 2.13 Session và Cookie trong PHP

Session và Cookie là hai cơ chế quan trọng trong việc quản lý trạng thái người dùng trong các ứng dụng web.

**Session** là cơ chế lưu trữ dữ liệu trên server, giúp duy trì thông tin người dùng trong suốt phiên làm việc của họ. Mỗi khi người dùng truy cập vào ứng dụng web, PHP sẽ tạo một session ID duy nhất để nhận diện người dùng. Thông tin của người dùng được lưu trữ trong server, giúp đảm bảo tính bảo mật và không bị thay đổi bởi người dùng. Session thường được sử dụng để lưu trữ các thông tin nhạy cảm như thông tin đăng nhập của người dùng, giỏ hàng trong các trang thương mại điện tử, v.v. Dữ liệu session có thể hết hạn sau một khoảng thời gian không hoạt động.

**Cookie** là cơ chế lưu trữ dữ liệu trên trình duyệt của người dùng. Các cookie có thể lưu trữ thông tin trên trình duyệt và được gửi đến server trong mỗi yêu cầu HTTP. Cookie thường được sử dụng để ghi nhớ các cài đặt hoặc tùy chọn của người dùng, ví dụ như ghi nhớ tài khoản đăng nhập hay lưu trữ các tùy chọn giao diện người dùng. Cookie có thể có thời gian sống ngắn (chỉ tồn tại trong phiên làm việc) hoặc lâu dài, tùy thuộc vào giá trị được thiết lập khi tạo cookie.

### Ưu điểm:

- **Session**: Quản lý trạng thái người dùng một cách bảo mật, dữ liệu không bị lộ ra ngoài.
- **Cookie**: Giúp lưu trữ thông tin trên trình duyệt, dễ dàng ghi nhớ thông tin giữa các phiên làm việc.

### Nhược điểm:

- **Session**: Dữ liệu được lưu trên server, có thể làm tăng tải cho server khi có nhiều người dùng.
- **Cookie**: Dữ liệu dễ bị sửa đổi hoặc xâm nhập nếu không mã hóa hợp lý, cũng như bị giới hạn về dung lượng.

Trong ứng dụng web, cả Session và Cookie đều đóng vai trò quan trọng trong việc duy trì trải nghiệm người dùng liên tục và an toàn, đặc biệt khi cần xử lý thông tin người dùng và các hành động không cần phải yêu cầu lại từ server.



## 2.14 Các lỗ hổng bảo mật phổ biến

Trong phát triển ứng dụng web, bảo mật luôn là một yếu tố quan trọng. Dưới đây là một số lỗ hổng bảo mật phổ biến:

**XSS (Cross-site scripting):** Lỗ hổng này xảy ra khi ứng dụng web cho phép người dùng nhập dữ liệu mà không kiểm tra hoặc làm sạch, khiến mã JavaScript độc hại có thể chạy trên trình duyệt của người dùng khác. Điều này có thể dẫn đến việc đánh cắp thông tin người dùng, thay đổi giao diện hoặc thậm chí kiểm soát tài khoản người dùng.

**SQL Injection:** Lỗ hổng này xảy ra khi ứng dụng không kiểm tra hoặc làm sạch các dữ liệu đầu vào của người dùng trước khi thực thi câu lệnh SQL. Điều này có thể cho phép kẻ tấn công thực thi các câu lệnh SQL độc hại, truy xuất hoặc thay đổi dữ liệu trong cơ sở dữ liệu.

**CSRF (Cross-site request forgery):** Đây là một loại tấn công mà kẻ tấn công dụ người dùng thực hiện một hành động không mong muốn trên ứng dụng mà người dùng đã đăng nhập, chẳng hạn như thay đổi thông tin tài khoản hoặc thực hiện thanh toán mà không có sự đồng ý của người dùng.

**Biện pháp phòng ngừa:**

- Sử dụng phương thức mã hóa và làm sạch dữ liệu người dùng.
- Kiểm tra tính hợp lệ của dữ liệu đầu vào và sử dụng prepared statements để tránh SQL injection.
- Sử dụng mã xác thực (tokens) để ngăn chặn CSRF.

## 2.15 SEO (Search Engine Optimization)

SEO (Search Engine Optimization) là quá trình tối ưu hóa trang web để cải thiện thứ hạng trên các công cụ tìm kiếm như Google, nhằm tăng cường khả năng hiển thị và thu hút người dùng. Việc tối ưu SEO giúp tăng lượng truy cập tự nhiên (organic traffic) mà không phải trả tiền cho quảng cáo.

**Các yếu tố quan trọng trong SEO:**

- Thẻ Meta: Thẻ meta title và meta description giúp cung cấp thông tin về nội dung trang web cho công cụ tìm kiếm và người dùng.
- URL thân thiện: Các URL cần dễ hiểu và chứa từ khóa quan trọng để công cụ tìm kiếm dễ dàng xác định nội dung trang.
- Tốc độ tải trang: Tốc độ tải trang nhanh là một yếu tố quan trọng để cải thiện trải nghiệm người dùng và xếp hạng trên công cụ tìm kiếm.
- Tối ưu hóa hình ảnh: Sử dụng thẻ alt cho hình ảnh để công cụ tìm kiếm có thể hiểu nội dung hình ảnh và cải thiện khả năng tìm kiếm.

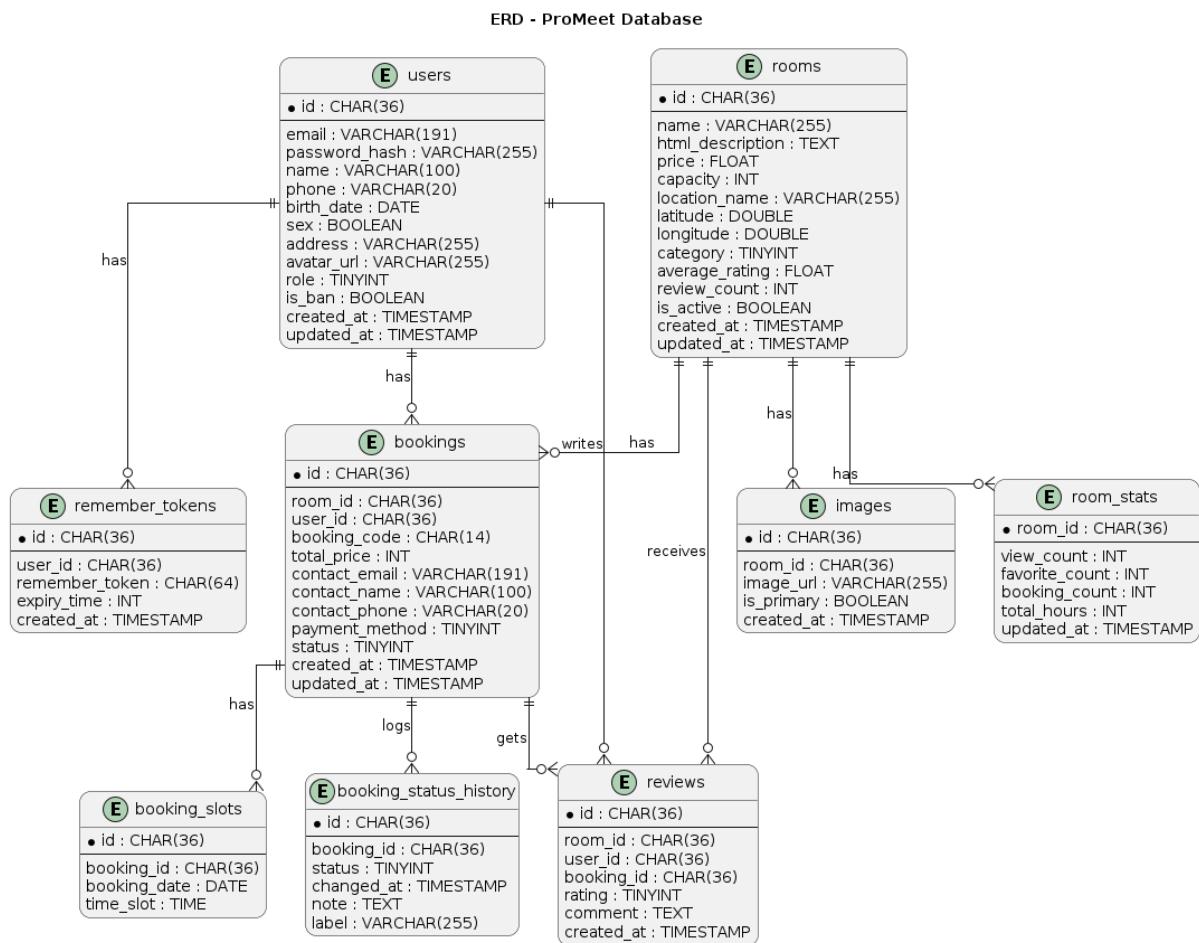
**Ưu điểm:**

- Giúp tăng lượng truy cập tự nhiên và cải thiện sự hiện diện của website trên các công cụ tìm kiếm.
- Làm cho website dễ dàng tiếp cận và thân thiện với người dùng.

# 3 Thiết kế ứng dụng

## 3.1 Thiết kế cơ sở dữ liệu

Dưới đây là lược đồ cơ sở dữ liệu ERD (Entity-Relation Diagram) cho toàn hệ thống ProMeet:



Hình 1: Lược đồ ERD của ProMeet

### 3.2 Mô tả chi tiết các thực thể

#### 3.2.1 Thực thể users

Thực thể **user** lưu trữ thông tin của người dùng trong hệ thống. Mỗi bản ghi đại diện cho một tài khoản người dùng duy nhất, bao gồm các thông tin nhận dạng như địa chỉ email, tên, số điện thoại, ngày sinh, và địa chỉ. Ngoài ra, bảng còn lưu trữ các thuộc tính bảo mật như mật khẩu đã mã hóa, trạng thái tài khoản (bị khóa hoặc không), vai trò người dùng (người dùng thông thường hoặc quản trị viên), cũng như các thông tin phụ như ảnh đại diện. Các trường `created_at` và `updated_at` giúp theo dõi thời điểm tạo và cập nhật gần nhất của bản ghi.



Thuộc tính	Kiểu dữ liệu	Mô tả
ID	CHAR(36)	Khóa chính
Email	VARCHAR(191)	Địa chỉ mail người dùng
Password_hash	VARCHAR(255)	Mật khẩu đã mã hóa
Name	VARCHAR(100)	Họ tên người dùng
Phone	VARCHAR(20)	Số điện thoại (duy nhất)
Birth_date	DATE	Ngày sinh
Sex	BOOLEAN	Giới tính (0: nam, 1: nữ)
Address	VARCHAR(255)	Địa chỉ người dùng
Avatar_url	VARCHAR(255)	Đường dẫn ảnh đại diện
Role	TINYINT	Vai trò (0: user, 1: admin)
Is_ban	BOOLEAN	Trạng thái khóa tài khoản (0: bình thường, 1: bị khóa)
Created_at	TIMESTAMP	Thời điểm tạo tài khoản
Updated_at	TIMESTAMP	Thời điểm cập nhật gần nhất

Bảng 1: Bảng mô tả chi tiết thực thể user

### 3.2.2 Thực thể rooms

Thực thể **rooms** lưu trữ thông tin các phòng họp có trong hệ thống. Mỗi bản ghi đại diện cho một phòng họp cụ thể mà người dùng có thể xem và đặt lịch. Thông tin của mỗi phòng bao gồm tên, mô tả chi tiết (dưới dạng HTML), mức giá thuê theo giờ, sức chứa tối đa, tên địa điểm, và vị trí địa lý thông qua tọa độ (latitude, longitude). Phòng còn được phân loại theo cấp độ (Basic, Standard, Premium) và lưu lại thông tin đánh giá trung bình cùng số lượng đánh giá. Trạng thái hoạt động của phòng (được phép hiển thị hay không) cũng được quản lý thông qua trường `is_active`. Hai trường `created_at` và `updated_at` dùng để theo dõi lịch sử thay đổi của phòng.

Thuộc tính	Kiểu dữ liệu	Mô tả
ID	CHAR(36)	Khóa chính
Name	VARCHAR(255)	Tên phòng họp
Html_description	TEXT	Mô tả chi tiết phòng dưới dạng HTML
Price	FLOAT UNSIGNED	Giá thuê phòng mỗi giờ
Capacity	INT UNSIGNED	Sức chứa tối đa (số người)
Location_name	VARCHAR(255)	Tên địa điểm hoặc chi nhánh
Latitude	DOUBLE	Vĩ độ của vị trí phòng
Longitude	DOUBLE	Kinh độ của vị trí phòng
Category	TINYINT	Phân loại phòng (0: Basic, 1: Standard, 2: Premium)
Average_rating	FLOAT UNSIGNED	Điểm đánh giá trung bình
Review_count	INT UNSIGNED	Số lượng đánh giá
Is_active	BOOLEAN	Trạng thái hoạt động (0: ẩn, 1: hiển thị)
Created_at	TIMESTAMP	Thời điểm tạo phòng
Updated_at	TIMESTAMP	Thời điểm cập nhật gần nhất

Bảng 2: Bảng mô tả chi tiết thực thể rooms



### 3.2.3 Thực thể bookings

Thực thể **bookings** lưu trữ thông tin các lượt đặt phòng họp của người dùng. Mỗi bản ghi đại diện cho một lượt đặt phòng cụ thể, liên kết giữa người dùng và phòng được đặt. Các thông tin chính bao gồm mã đặt phòng (**booking\_code**), tổng chi phí thanh toán, thông tin liên hệ (tên, email, số điện thoại), và phương thức thanh toán. Trạng thái của mỗi đơn đặt phòng được biểu diễn qua các giá trị số, giúp hệ thống dễ dàng quản lý tiến trình đặt phòng từ khi khởi tạo đến khi hoàn tất hoặc hủy bỏ. Trường **created\_at** và **updated\_at** hỗ trợ theo dõi quá trình thay đổi đơn đặt.

Thuộc tính	Kiểu dữ liệu	Mô tả
ID	CHAR(36)	Khóa chính
Room_id	CHAR(36)	Khóa ngoại đến phòng được đặt
User_id	CHAR(36)	Khóa ngoại đến người đặt phòng
Booking_code	CHAR(14)	Mã đặt phòng (ngẫu nhiên, không trùng)
Total_price	INT UNSIGNED	Tổng số tiền cần thanh toán
Contact_email	VARCHAR(191)	Email người liên hệ khi đặt
Contact_name	VARCHAR(100)	Tên người liên hệ khi đặt
Contact_phone	VARCHAR(20)	Số điện thoại liên hệ
Payment_method	TINYINT	Phương thức thanh toán (0: ngân hàng, 1: MoMo)
Status	TINYINT	Trạng thái đặt phòng (0: chờ xử lý, 1: đã thanh toán, 2: đã xác nhận, 3: hoàn tất, 4: đã hủy)
Created_at	TIMESTAMP	Thời điểm tạo đơn đặt
Updated_at	TIMESTAMP	Thời điểm cập nhật gần nhất

Bảng 3: Bảng mô tả chi tiết thực thể bookings

### 3.2.4 Thực thể booking\_slots

Thực thể **booking\_slots** lưu thông tin chi tiết về các khung giờ mà người dùng đã đặt trong mỗi lượt đặt phòng. Một đơn đặt phòng (**booking**) có thể bao gồm nhiều khung giờ đặt khác nhau. Điều này cho phép hệ thống hỗ trợ đặt phòng theo đơn vị nhỏ (như 30 phút, 1 giờ). Mỗi bản ghi trong bảng này biểu diễn một slot cụ thể được đặt vào một ngày và giờ nhất định, gắn với một **booking\_id**.

Thuộc tính	Kiểu dữ liệu	Mô tả
ID	CHAR(36)	Khóa chính
Booking_id	CHAR(36)	Khóa ngoại liên kết với bảng bookings
Booking_date	DATE	Ngày đặt phòng
Time_slot	TIME	Giờ bắt đầu của khung giờ đặt

Bảng 4: Bảng mô tả chi tiết thực thể booking\_slots

### 3.2.5 Thực thể booking\_status\_history

Thực thể **booking\_status\_history** lưu lại lịch sử thay đổi trạng thái của mỗi đơn đặt phòng. Mỗi khi trạng thái của một đơn thay đổi (ví dụ: từ pending sang paid), một bản ghi mới sẽ được tạo ra để ghi nhận trạng thái mới, thời điểm thay đổi, và ghi chú kèm theo nếu có. Điều này hỗ trợ theo dõi quá trình xử lý đơn hàng và phục vụ mục đích kiểm tra, thống kê hoặc hỗ trợ khách hàng.



Thuộc tính	Kiểu dữ liệu	Mô tả
ID	CHAR(36)	Khóa chính
Booking_id	CHAR(36)	Khóa ngoại liên kết với bảng bookings
Status	TINYINT	Trạng thái đơn (0: chờ xử lý, 1: đã thanh toán, 2: đã xác nhận, 3: hoàn tất, 4: đã hủy)
Changed_at	TIMESTAMP	Thời điểm thay đổi trạng thái
Note	TEXT	Ghi chú liên quan đến thay đổi trạng thái
Label	VARCHAR(255)	Nhãn tùy chỉnh cho trạng thái (nếu có)

Bảng 5: Bảng mô tả chi tiết thực thể booking\_status\_history

### 3.2.6 Thực thể reviews

Thực thể **reviews** lưu trữ đánh giá của người dùng về các phòng họp sau khi đã hoàn tất việc đặt phòng. Mỗi bản ghi chứa thông tin về người đánh giá, phòng được đánh giá, đơn đặt phòng liên quan, điểm đánh giá (từ 1 đến 5 sao), nội dung nhận xét, và thời điểm tạo. Việc gắn kết đánh giá với **booking\_id** giúp đảm bảo rằng chỉ người dùng đã từng đặt phòng mới có thể đánh giá phòng đó.

Thuộc tính	Kiểu dữ liệu	Mô tả
ID	CHAR(36)	Khóa chính
Room_id	CHAR(36)	Khóa ngoại liên kết với bảng rooms
User_id	CHAR(36)	Khóa ngoại liên kết với bảng users
Booking_id	CHAR(36)	Khóa ngoại liên kết với bảng bookings
Rating	TINYINT	Điểm đánh giá (1 đến 5 sao)
Comment	TEXT	Nội dung đánh giá của người dùng
Created_at	TIMESTAMP	Thời điểm tạo đánh giá

Bảng 6: Bảng mô tả chi tiết thực thể reviews

### 3.2.7 Thực thể images

Thực thể **images** lưu trữ các hình ảnh liên quan đến phòng họp. Mỗi bản ghi chứa một đường dẫn hình ảnh cùng với thông tin phòng họp tương ứng. Thuộc tính **is\_primary** xác định xem hình ảnh có phải là ảnh đại diện chính của phòng hay không. Một phòng có thể có nhiều hình ảnh, nhưng chỉ một hình ảnh được đánh dấu là chính.

Thuộc tính	Kiểu dữ liệu	Mô tả
ID	CHAR(36)	Khóa chính
Room_id	CHAR(36)	Khóa ngoại liên kết với bảng rooms
Image_url	VARCHAR(255)	Đường dẫn đến ảnh phòng họp
Is_primary	BOOLEAN	Có phải ảnh đại diện chính hay không (0: không, 1: có)
Created_at	TIMESTAMP	Thời điểm tạo ảnh

Bảng 7: Bảng mô tả chi tiết thực thể images



### 3.2.8 Thực thể remember\_tokens

Thực thể **remember\_tokens** lưu trữ các mã "ghi nhớ đăng nhập" (remember me) được tạo ra khi người dùng chọn tùy chọn ghi nhớ tài khoản. Mỗi token được liên kết với một người dùng cụ thể và có thời gian hết hạn. Khi người dùng quay lại trình duyệt trong thời gian hiệu lực, hệ thống có thể tự động xác thực lại bằng mã này.

Thuộc tính	Kiểu dữ liệu	Mô tả
ID	CHAR(36)	Khóa chính
User_id	CHAR(36)	Khóa ngoại đến bảng users
Remember_token	CHAR(64)	Mã ghi nhớ đăng nhập
Expiry_time	INT	Thời gian hết hạn của token (đơn vị: giây từ Epoch)
Created_at	TIMESTAMP	Thời điểm tạo mã token

Bảng 8: Bảng mô tả chi tiết thực thể remember\_tokens

### 3.2.9 Thực thể room\_stats

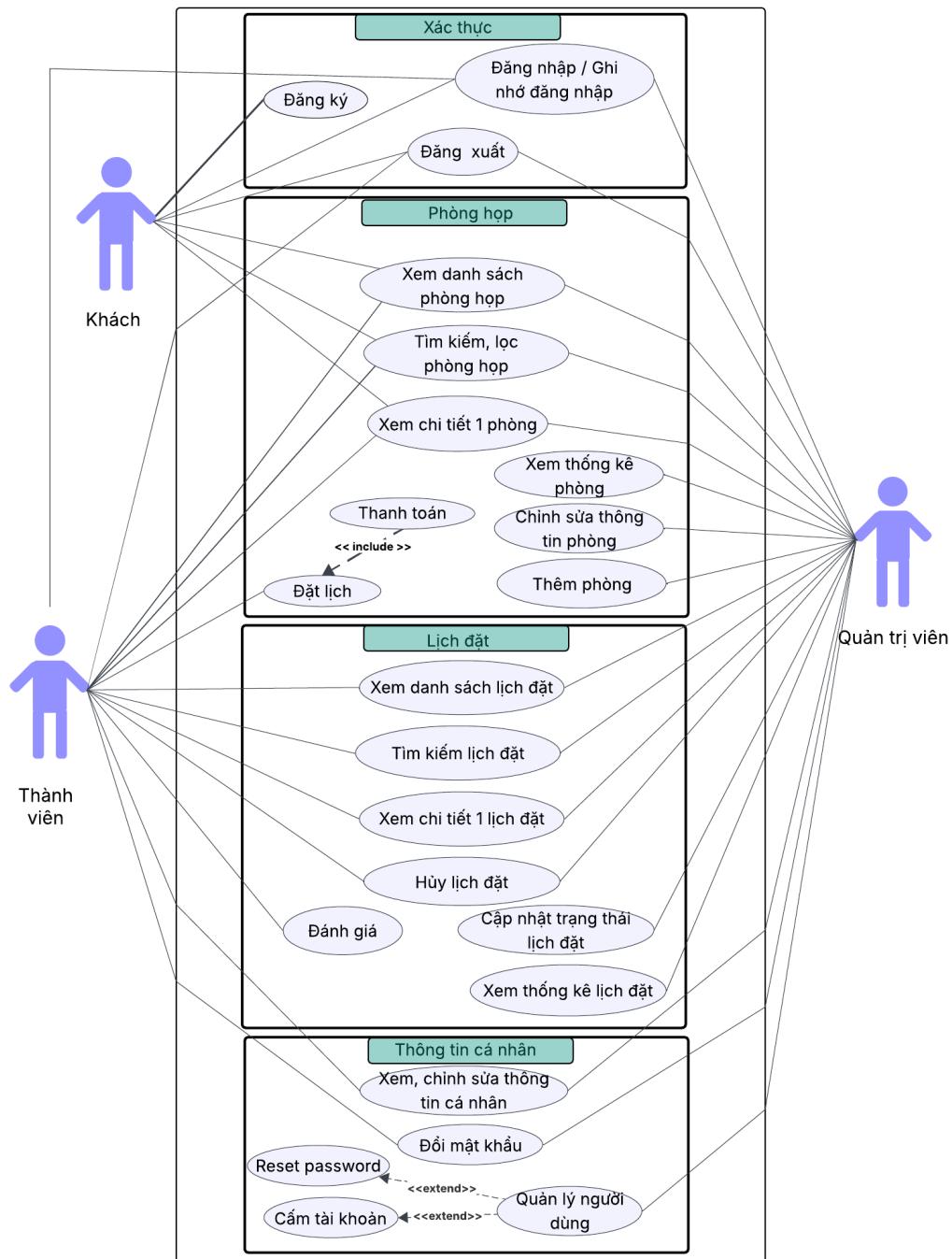
Thực thể **room\_stats** lưu trữ các thông tin kê liên quan đến phòng họp. Các thông kê bao gồm số lượt xem, số lượt yêu thích, số lượt đặt phòng và tổng số giờ đã được đặt phòng. Thực thể này giúp theo dõi hoạt động của từng phòng họp, hỗ trợ việc phân tích và tối ưu hóa việc cung cấp dịch vụ.

Thuộc tính	Kiểu dữ liệu	Mô tả
Room_id	CHAR(36)	Khóa chính liên kết với bảng rooms
View_count	INT UNSIGNED	Số lượt xem phòng
Favorite_count	INT UNSIGNED	Số lượt yêu thích phòng
Booking_count	INT UNSIGNED	Số lượt đặt phòng
Total_hours	INT UNSIGNED	Tổng số giờ đã đặt phòng
Updated_at	TIMESTAMP	Thời điểm cập nhật thông tin thống kê

Bảng 9: Bảng mô tả chi tiết thực thể room\_stats

## 3.3 Sơ đồ Usecase Diagram

Dưới đây là sơ đồ usecase cho toàn hệ thống ProMeet:



Hình 2: Sơ đồ Usecase Diagram của ProMeet



### 3.4 Đặc tả Usecase

#### 3.4.1 <User> Đặt phòng họp (UC\_01)

Use-case name	Đặt phòng họp		
Created by	Trần Quang Tác	Last updated by	Trần Quang Tác
Date created	04/05/2025	Date last updated	05/05/2025
Actors	Người dùng		
Description	Người dùng có thể tìm và đặt phòng họp phù hợp với nhu cầu, thông qua việc chọn thời gian, nhập thông tin liên hệ và tiến hành thanh toán.		
Preconditions	Người dùng phải đăng nhập vào hệ thống.		
Postconditions	<ol style="list-style-type: none"><li>Hệ thống tạo đơn đặt lịch thành công cho người dùng.</li><li>Hệ thống sẽ cập nhật thông tin đơn hàng trong cơ sở dữ liệu.</li></ol>		
Normal flows	<ol style="list-style-type: none"><li>Người dùng truy cập danh sách phòng họp.</li><li>Người dùng chọn một phòng họp mong muốn.</li><li>Nhấn nút "Đặt lịch ngay" để cuộn đến phần đặt lịch.</li><li>Hệ thống hiển thị các khung giờ khả dụng theo block 30 phút.</li><li>Người dùng chọn khung giờ phù hợp.</li><li>Hệ thống hiển thị tổng tiền và yêu cầu xác nhận đặt lịch.</li><li>Sau khi xác nhận, hệ thống tạo đơn đặt phòng ở trạng thái 'Chờ thanh toán', chuyển sang trang giữ chỗ.</li><li>Người dùng nhập thông tin liên hệ (tên, SDT, email).</li><li>Người dùng có tối đa 10 phút để thanh toán (MoMo, ngân hàng, v.v).</li><li>Nếu thanh toán thành công, trạng thái đơn chuyển thành 'Đã thanh toán' và sau đó là 'Chờ xác nhận'.</li><li>Admin kiểm tra và xác nhận đơn – trạng thái chuyển thành 'Đã xác nhận'.</li></ol>		
Exceptions	<ol style="list-style-type: none"><li>Nếu hết thời gian giữ chỗ mà chưa thanh toán, đơn bị huỷ.</li><li>Nếu khung giờ được người khác đặt trước khi hoàn tất, hệ thống thông báo và yêu cầu chọn lại.</li></ol>		
Note and issues	<ol style="list-style-type: none"><li>Mỗi khung giờ phải cách nhau ít nhất 30 phút để đảm bảo thời gian dọn phòng.</li><li>Người dùng được phép huỷ đơn nếu chưa đến thời gian sử dụng.</li></ol>		

Bảng 10: Bảng mô tả use case Đặt phòng họp



### 3.4.2 <User> Chính sửa thông tin cá nhân (UC\_02))

Use-case name	Chỉnh sửa thông tin cá nhân		
Created by	Trần Quang Tác	Last updated by	Trần Quang Tác
Date created	04/05/2025	Date last updated	05/05/2025
Actors	Người dùng		
Description	Người dùng có thể chỉnh sửa thông tin cá nhân như họ tên, số điện thoại, ảnh đại diện,... để cập nhật hồ sơ cá nhân của mình.		
Preconditions	Người dùng phải đăng nhập vào hệ thống.		
Postconditions	<ol style="list-style-type: none"><li>Thông tin cá nhân của người dùng được cập nhật trong hệ thống.</li><li>Hệ thống phản hồi thành công nếu cập nhật hợp lệ.</li></ol>		
Normal flows	<ol style="list-style-type: none"><li>Người dùng truy cập trang "Hồ sơ".</li><li>Nhấn vào nút "Chỉnh sửa hồ sơ".</li><li>Hệ thống hiển thị form chứa các trường thông tin: họ tên, số điện thoại, ảnh đại diện, v.v.</li><li>Người dùng cập nhật các trường mong muốn.</li><li>Nhấn nút "Lưu hồ sơ".</li><li>Hệ thống kiểm tra tính hợp lệ của thông tin.</li><li>Nếu hợp lệ, thông tin được cập nhật vào cơ sở dữ liệu và hiển thị thông báo thành công.</li></ol>		
Exceptions	<ol style="list-style-type: none"><li>Nếu trường nhập không hợp lệ (ví dụ số điện thoại sai định dạng), hệ thống hiển thị lỗi.</li><li>Nếu cập nhật thất bại do lỗi hệ thống, hiển thị thông báo lỗi và giữ nguyên dữ liệu cũ.</li></ol>		
Note and issues	<ol style="list-style-type: none"><li>Người dùng không được phép thay đổi địa chỉ email.</li><li>Ảnh đại diện cần giới hạn dung lượng và định dạng (JPEG/PNG).</li></ol>		

Bảng 11: Bảng mô tả use case Chính sửa thông tin cá nhân



### 3.4.3 <Admin> Thêm phòng họp mới (UC\_01))

Use-case name	Thêm phòng		
Created by	Trần Quang Tác	Last updated by	Trần Quang Tác
Date created	05/05/2025	Date last updated	07/05/2025
Actors	Admin		
Description	Admin có thể thêm mới một phòng họp vào hệ thống để người dùng có thể đặt lịch.		
Preconditions	Admin đã đăng nhập vào hệ thống với quyền quản trị viên.		
Postconditions	1. Phòng họp mới được lưu vào hệ thống. 2. Admin có thể cập nhật thêm thông tin chi tiết và kích hoạt phòng sau đó.		
Normal flows	1. Admin bấm vào trang "Phòng họp". 2. Nhấn vào nút "Thêm phòng". 3. Hệ thống hiển thị form nhập thông tin: tên phòng, giá theo giờ, địa điểm, loại phòng, sức chứa, kinh độ, vĩ độ. 4. Admin điền thông tin và bấm "Thêm phòng". 5. Hệ thống kiểm tra tính hợp lệ. 6. Nếu hợp lệ: lưu phòng vào CSDL, hiển thị thông báo thành công và điều hướng tới trang chi tiết phòng. 7. Tại trang chi tiết, admin có thể cập nhật ảnh, mô tả,... và bấm "Kích hoạt phòng".		
Exceptions	Nếu thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu chỉnh sửa.		
Note and issues	Nên kiểm tra xem tên phòng có bị trùng trong cùng địa điểm hay không.		

Bảng 12: Bảng mô tả use case Thêm phòng họp mới



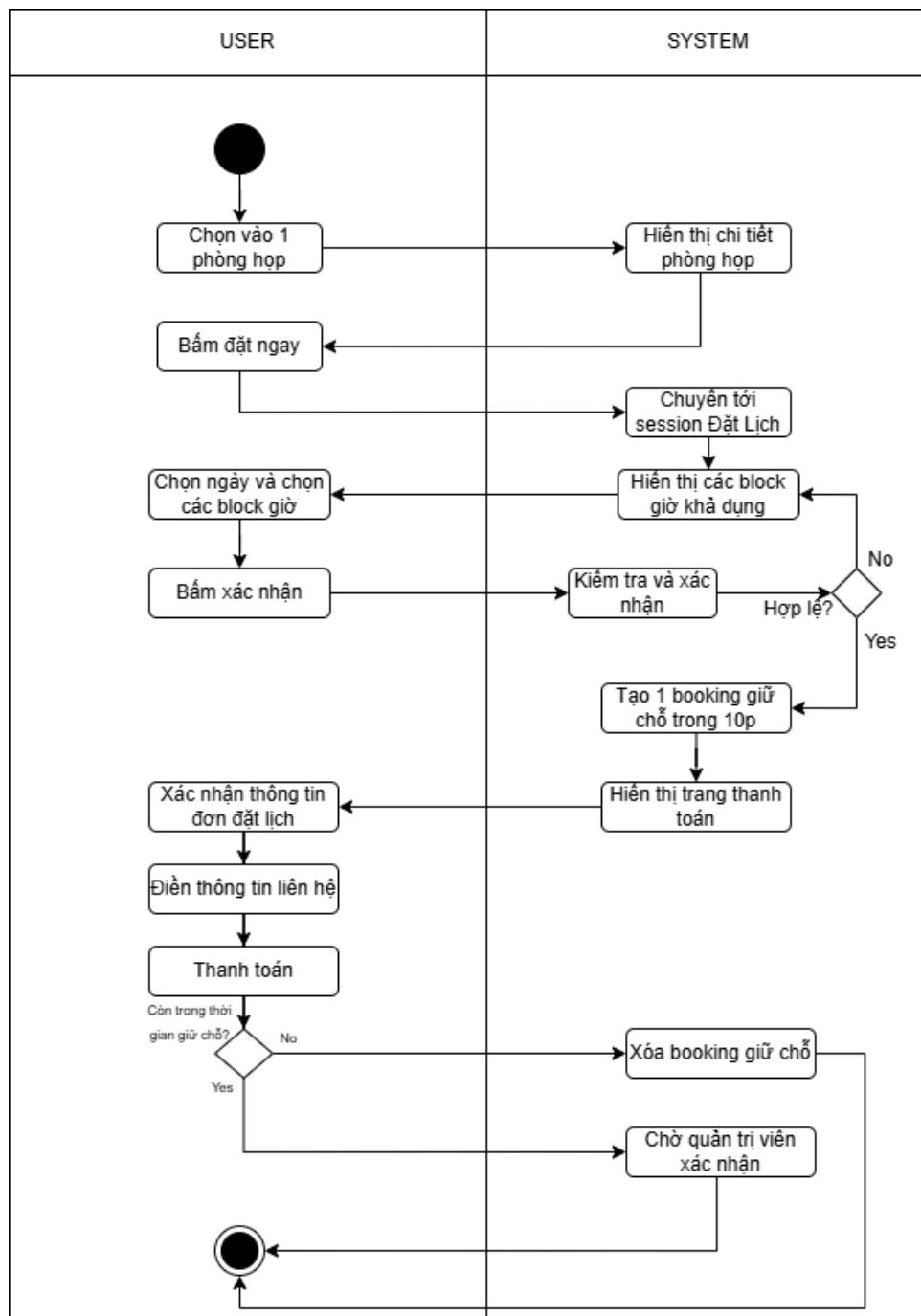
### 3.4.4 <Admin> Cập nhật trạng thái lịch đặt (UC\_02))

Use-case name	Cập nhật trạng thái lịch đặt		
Created by	Trần Quang Tác	Last updated by	Trần Quang Tác
Date created	05/05/2025	Date last updated	05/05/2025
Actors	Admin		
Description	Admin có thể thay đổi trạng thái của các đơn đặt phòng sau khi người dùng đã thanh toán hoặc có các thay đổi phát sinh.		
Preconditions	Admin đã đăng nhập và truy cập vào trang quản lý lịch đặt.		
Postconditions	1. Trạng thái đơn đặt phòng được cập nhật trong hệ thống. 2. Người dùng sẽ nhận được trạng thái mới của đơn hàng.		
Normal flows	1. Admin truy cập vào mục “Lịch đặt”. 2. Admin tìm và chọn đơn đặt phòng cần cập nhật. 3. Hệ thống hiển thị thông tin chi tiết đơn đặt phòng. 4. Admin chọn trạng thái cần cập nhật: “Xác nhận”, “Hủy”, “Hoàn tất”,... 5. Admin xác nhận thay đổi. 6. Hệ thống kiểm tra và cập nhật trạng thái vào cơ sở dữ liệu. 7. Hệ thống hiển thị thông báo thành công.		
Exceptions	1. Nếu trạng thái không hợp lệ (VD: cập nhật sang “Hoàn tất” khi chưa “Xác nhận”), hệ thống báo lỗi.		
Note and issues	Nên ghi log các thay đổi trạng thái phục vụ kiểm tra sau này.		

Bảng 13: Bảng mô tả use case Cập nhật trạng thái lịch đặt

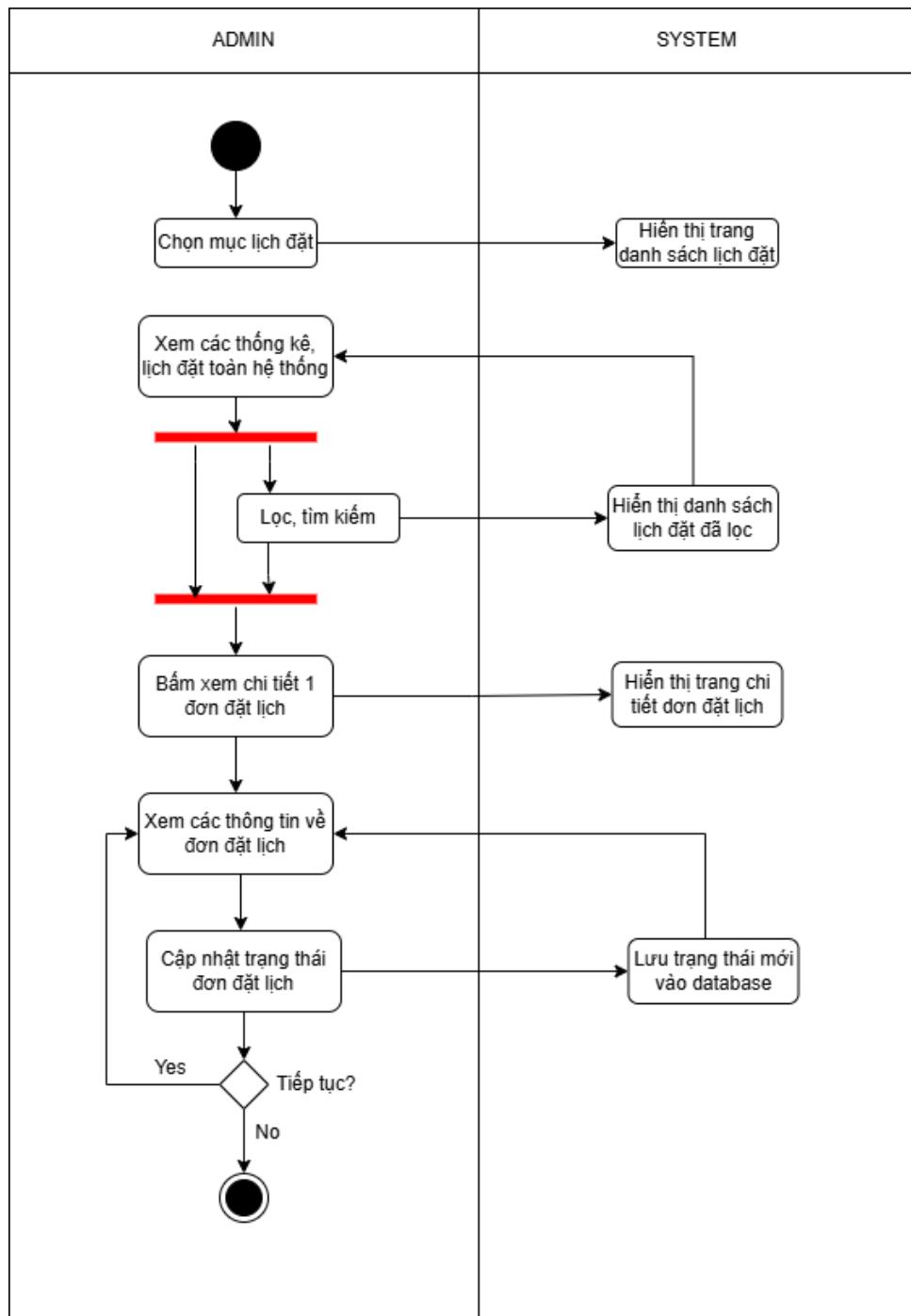
### 3.5 Activity Diagram

#### 3.5.1 Đặt phòng họp (User)



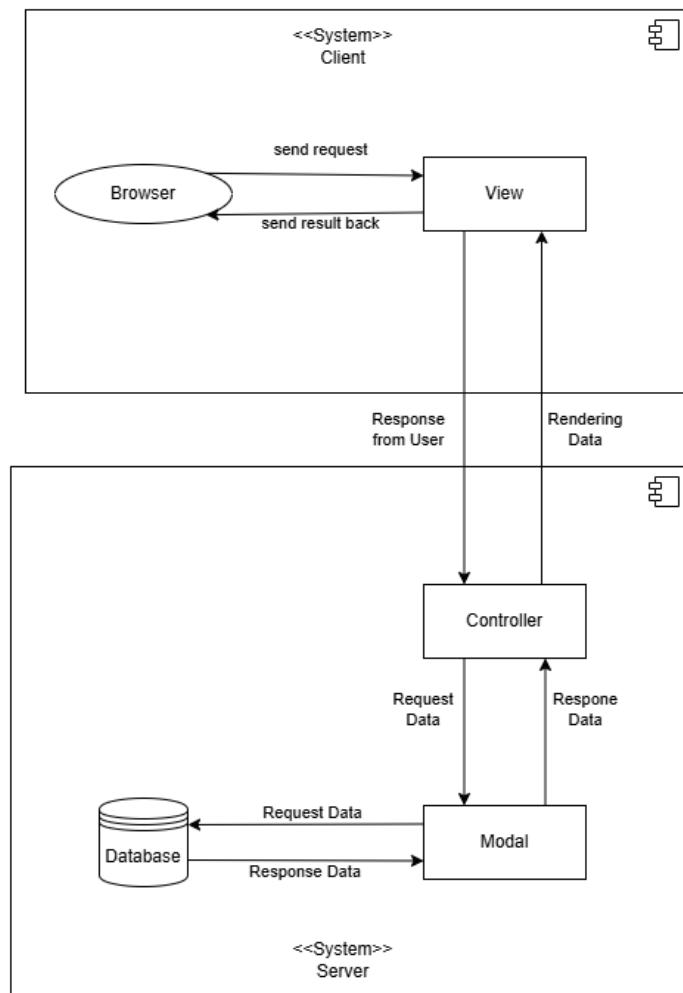
Hình 3: Activity diagram Đặt phòng họp (User)

### 3.5.2 Duyệt / Quản lý đơn đặt phòng (Admin)



Hình 4: Activity diagram Duyệt / Quản lý đơn đặt phòng (Admin)

### 3.6 Kiến trúc hệ thống



Hình 5: Architecture Diagram

Kiến trúc hệ thống ProMeet được xây dựng theo mô hình **MVC (Model-View-Controller)**, giúp tách biệt rõ ràng các phần xử lý logic, giao diện người dùng và dữ liệu. Mô hình MVC không chỉ giúp tổ chức mã nguồn rõ ràng mà còn dễ dàng mở rộng và bảo trì hệ thống theo thời gian. Cấu trúc này hỗ trợ khả năng mở rộng và linh hoạt, giúp việc phát triển các tính năng mới hoặc sửa lỗi trở nên đơn giản hơn.

#### 1. Model (Mô hình)

Phần **Model** trong kiến trúc MVC của ProMeet đảm nhiệm việc quản lý dữ liệu, logic nghiệp vụ và trạng thái của hệ thống. Các đối tượng mô hình tương tác trực tiếp với cơ sở dữ liệu, thực hiện các thao tác CRUD (Create, Read, Update, Delete). Model không phụ thuộc vào giao diện người dùng và chỉ tập trung vào việc xử lý dữ liệu.

**Ví dụ về các lớp Model trong ProMeet:**

- **RoomModel:** Quản lý dữ liệu phòng họp (như tên phòng, loại phòng, giá phòng, số lượng người tối đa, và trạng thái của phòng).
- **UserModel:** Quản lý thông tin người dùng (như email, mật khẩu, thông tin cá nhân).



- **BookingModel:** Quản lý các đơn đặt phòng, bao gồm thông tin người dùng, phòng họp đã đặt, trạng thái đơn và thời gian.
- **ReviewModel:** Quản lý các đánh giá của người dùng về phòng họp.

## 2. View (Giao diện)

Phần **View** là thành phần xử lý giao diện người dùng. Nó chịu trách nhiệm hiển thị dữ liệu từ **Model** và nhận các đầu vào từ người dùng. Trong ProMeet, View bao gồm tất cả các trang web và giao diện người dùng, từ màn hình đăng nhập cho đến trang danh sách phòng họp và trang chi tiết phòng. Các **View** này được tạo ra bằng HTML kết hợp với các tệp CSS, JavaScript để làm giao diện động và tương tác với người dùng.

**Ví dụ về các View trong ProMeet:**

- **Trang đăng nhập:** Hiển thị form đăng nhập với các trường nhập liệu cho email và mật khẩu.
- **Trang danh sách phòng:** Hiển thị các phòng họp có sẵn với thông tin cơ bản (tên phòng, giá phòng, số lượng người, đánh giá...).
- **Trang chi tiết phòng:** Hiển thị các thông tin chi tiết về phòng, các đánh giá người dùng, hình ảnh phòng và khả năng đặt phòng.
- **Trang quản trị:** Dành cho các quản trị viên để quản lý người dùng, phòng họp, và đơn hàng.

## 3. Controller (Điều khiển)

Phần **Controller** chịu trách nhiệm xử lý các yêu cầu từ người dùng (requests), tương tác với các **Model** để lấy dữ liệu và cuối cùng trả về **View** phù hợp. Controller là cầu nối giữa **Model** và **View**, đảm bảo rằng giao diện người dùng được cập nhật chính xác dựa trên các thao tác và trạng thái của hệ thống.

**Ví dụ về các Controller trong ProMeet:**

- **RoomController:** Xử lý các yêu cầu liên quan đến phòng họp, như liệt kê phòng, tìm kiếm phòng, và hiển thị chi tiết phòng.
- **BookingController:** Quản lý các thao tác đặt phòng, như đặt phòng, xác nhận thanh toán, và cập nhật trạng thái đơn.
- **UserController:** Xử lý các chức năng liên quan đến người dùng, như đăng nhập, đăng ký, cập nhật thông tin cá nhân, và quản lý tài khoản.
- **ReviewController:** Quản lý các thao tác liên quan đến đánh giá phòng họp, bao gồm việc đăng và hiển thị các đánh giá của người dùng.

## 4. Quy trình hoạt động

Quy trình hoạt động của hệ thống **ProMeet** theo mô hình MVC có thể tóm tắt như sau:

1. **Yêu cầu từ người dùng:** Khi người dùng gửi yêu cầu (ví dụ, yêu cầu đặt phòng), yêu cầu này sẽ được gửi tới **Controller**.
2. **Xử lý yêu cầu:** **Controller** nhận yêu cầu và tương tác với **Model** để thực hiện các thao tác cần thiết (chẳng hạn như lấy dữ liệu phòng họp, cập nhật đơn đặt phòng).



3. **Cập nhật View:** Sau khi có dữ liệu từ **Model, Controller** sẽ truyền dữ liệu này đến **View**, để giao diện người dùng được cập nhật phù hợp với yêu cầu và phản hồi từ hệ thống.
4. **Trả kết quả cho người dùng:** Giao diện người dùng (View) hiển thị kết quả (ví dụ, danh sách phòng họp, chi tiết phòng, thông báo đặt phòng thành công, v.v.).

## 5. Cấu trúc thư mục của ProMeet

Mô hình MVC được tổ chức theo cấu trúc thư mục rõ ràng như sau:

```
/app
    /Controllers      <-- Xử lý các yêu cầu HTTP và điều hướng đến mô hình.
    /Models          <-- Chứa các lớp mô hình tương tác với cơ sở dữ liệu.
    /views           <-- Các file giao diện.
    /Core             <-- Các thành phần cốt lõi như Database, Logger, Router.

/public
    /assets          <-- Các file tĩnh: CSS, JS, hình ảnh.
    /mazer           <-- Theme Mazer.
    /upload          <-- File tải lên từ người dùng.
    /.htaccess       <-- Cấu hình rewrite URL và bảo vệ thư mục nội bộ
    /index.php        <-- Điểm vào của ứng dụng.

/config
    /config.php     <-- Cấu hình hệ thống (Database, etc.).
```

```
/logs
    /app_error.log  <-- Ghi lại lỗi ứng dụng phục vụ debug.
```

## 3.7 Design Pattern

Dự án ProMeet sử dụng một số mẫu thiết kế (design pattern) phổ biến để đảm bảo tính mở rộng, dễ bảo trì và tái sử dụng mã nguồn:

- **Singleton Pattern:** Được áp dụng trong lớp **Database, LogService** và **Container** nhằm đảm bảo chỉ khởi tạo một instance duy nhất trong toàn hệ thống. Điều này giúp tiết kiệm tài nguyên, đặc biệt là với kết nối cơ sở dữ liệu hoặc hệ thống ghi log.



```
class Database
{
    private $pdo;
    private static $instance = null;

    private function __construct()
    {
        $this->connect();
    }

    // Singleton pattern để tạo ra 1 instance duy nhất
    public static function getInstance(): Database
    {
        if (self::$instance === null) {
            self::$instance = new Database();
        }

        return self::$instance;
    }
}
```

Hình 6: Singleton trong lớp Database

```
class LogService
{
    private static $instance = null;
    private $logDirectory;

    private function __construct($logDirectory = '../logs')
    {
        $this->logDirectory = $logDirectory;
        if (!file_exists($this->logDirectory)) {
            mkdir($this->logDirectory, 0777, true);
        }
    }

    // Phương thức static để lấy instance duy nhất
    public static function getInstance($logDirectory = '../logs')
    {
        if (self::$instance === null) {
            self::$instance = new LogService($logDirectory);
        }
        return self::$instance;
    }
}
```

Hình 7: Singleton trong lớp LogService

```
class Container
{
    protected static $instance = null;
    protected $bindings = [];

    public static function getInstance(): self
    {
        if (!self::$instance) {
            self::$instance = new self();
        }
        return self::$instance;
    }
}
```

Hình 8: Singleton trong lớp Container

**Ví dụ sử dụng:** Việc khởi tạo và sử dụng các singleton được thực hiện thông qua Container:



```
$container->set('db', function () {
    return Database::getInstance(); // Singleton
});

$container->set('logger', function () {
    return LogService::getInstance(); // Singleton
});
```

Hình 9: Sử dụng các Singleton trong Container

- **Service Container / Dependency Injection:** Thông qua lớp Container, các thành phần như Database, Logger, Model được đăng ký và quản lý tập trung. Việc lấy các dịch vụ thông qua `$container->get()` giúp tách biệt phụ thuộc và hỗ trợ kiểm thử (testability) tốt hơn.

```
class Container
{
    protected static $instance = null;
    protected $bindings = [];

    public static function getInstance(): self
    {
        if (!self::$instance) {
            self::$instance = new self();
        }
        return self::$instance;
    }

    public function set(string $key, callable $resolver)
    {
        $this->bindings[$key] = $resolver;
    }

    public function get(string $key)
    {
        if (!isset($this->bindings[$key])) {
            throw new \Exception("Service {$key} not found in container.");
        }
        return $this->bindings[$key]($this);
    }
}
```

Hình 10: Các hàm `get` và `set` trong lớp Container

```
class HomeController {
    protected $log;
    protected $userModel;
    protected $bookingModel;
    protected $reviewModel;
    protected $roomModel;

    public function __construct(Container $container) {
        $this->log = $container->get('logger');
        $this->userModel = $container->get('UserModel');
        $this->bookingModel = $container->get('BookingModel');
        $this->reviewModel = $container->get('ReviewModel');
        $this->roomModel = $container->get('RoomModel');
    }
}
```

Hình 11: Ví dụ sử dụng `get` để lấy dịch vụ từ Container

```
// Đăng ký các model
$container->set('UserModel', function ($c) {
    return new UserModel(
        $c->get('db'),
        $c->get('logger')
    );
});
```

Hình 12: Ví dụ sử dụng `set` để đăng ký dịch vụ vào Container



## 4 Hiện thực

### 4.1 Về phía khách (chưa đăng nhập)

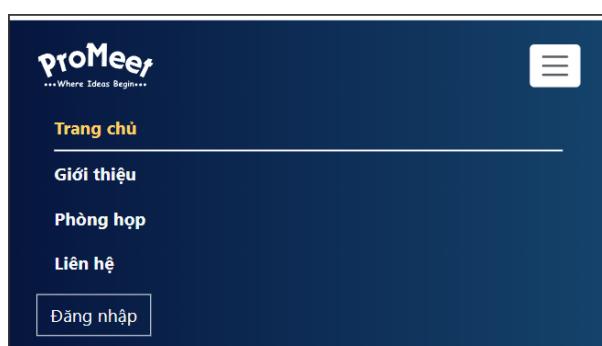
Các trang giao diện sẽ có navbar và footer chung.



Hình 13: Thanh navbar phía người dùng khách



Hình 14: Thanh navbar phía người dùng khách ở chế độ nhỏ



Hình 15: Thanh navbar phía người dùng khách ở chế độ nhỏ sau khi bấm mở rộng



Hình 16: Phần footer phía người dùng khách



Phần đầu của trang chủ là **carousel giới thiệu**, giúp tạo ấn tượng ban đầu với người dùng. Carousel trình bày các thông điệp nổi bật của nền tảng, kết hợp hình ảnh minh họa hiện đại và nút kêu gọi hành động như “Khám phá ngay”, giúp người dùng dễ dàng tiếp cận dịch vụ.



Hình 17: Phần đầu trang chủ phía người dùng khách

Tiếp theo là phần **giới thiệu về ProMeet**, trình bày thông tin tổng quan về nền tảng, kèm theo phần mô tả **sứ mệnh và tầm nhìn** nhằm khẳng định định hướng phát triển và giá trị mà ProMeet mang lại cho người dùng.



**ProMeet**  
...Where Ideas Begin...

Trang chủ    **Giới thiệu**    Phòng họp    Liên hệ    Đăng nhập

## Về ProMeet



ProMeet là nền tảng giúp bạn đặt phòng họp một cách nhanh chóng, tiết kiệm thời gian và chi phí. Với hệ thống tìm kiếm thông minh và giao diện thân thiện, bạn sẽ dễ dàng tìm được không gian phù hợp cho mọi buổi họp, sự kiện hoặc đào tạo.

Hình 18: Giới thiệu tổng quan về ProMeet

**ProMeet**  
...Where Ideas Begin...

Trang chủ    **Giới thiệu**    Phòng họp    Liên hệ    Đăng nhập

## Sứ mệnh & Tâm nhìn

Chúng tôi tin rằng việc tìm kiếm một không gian họp lý tưởng không nên là điều khó khăn. ProMeet cam kết xây dựng một hệ sinh thái đặt phòng chuyên nghiệp, minh bạch và hiệu quả, giúp kết nối các doanh nghiệp, cá nhân và tổ chức một cách nhanh chóng và tiện lợi nhất.

**Hiệu quả & Tiện lợi**  
Tim kiếm và đặt phòng chỉ trong vài bước, giúp bạn tiết kiệm thời gian cho những việc quan trọng hơn.

**Minh bạch & Uy tín**  
Thông tin rõ ràng, giá cả công khai, hệ thống đánh giá xác thực giúp bạn yên tâm khi lựa chọn.

**Đồng hành & Phát triển**  
Không ngừng cải tiến sản phẩm để phục vụ tốt hơn, hỗ trợ các đối tác khai thác tối đa công suất phòng họp.

Hình 19: Sứ mệnh và tầm nhìn của ProMeet

Tiếp đến là phần Phòng họp nổi bật, nơi các phòng họp đặc biệt của ProMeet được giới thiệu nổi bật trên trang chủ.



The screenshot shows the 'Phòng họp nổi bật' (Notable Meeting Rooms) section of the ProMeet website. At the top, there is a navigation bar with links: Trang chủ (Home), Giới thiệu (Introduction), Phòng họp (Meeting Room), Liên hệ (Contact), and Đăng nhập (Login). Below the navigation, the title 'Phòng họp nổi bật' is displayed with a blue underline. To the right of the title is a button labeled 'Xem tất cả phòng →'. There are three cards, each featuring a photo of a meeting room and details about it:

- Phòng họp Ánh Dương**  
Thành phố Thủ Đức  
[Xem chi tiết](#)
- Phòng họp Sáng Tạo**  
Quận 1  
[Xem chi tiết](#)
- Phòng họp Kết Nối**  
Quận 10  
[Xem chi tiết](#)

Hình 20: Phòng họp nổi bật trên trang chủ phía người dùng khách

Tiếp đến là phần Liên hệ, nơi người dùng có thể tìm thấy thông tin để liên lạc với ProMeet, hỗ trợ các yêu cầu và thắc mắc.

The screenshot shows the 'Liên hệ với chúng tôi' (Contact Us) section of the ProMeet website. At the top, there is a navigation bar with links: Trang chủ (Home), Giới thiệu (Introduction), Phòng họp (Meeting Room), Liên hệ (Contact), and Đăng nhập (Login). Below the navigation, the title 'Liên hệ với chúng tôi' is displayed with a blue underline. A message below the title reads: 'Chúng tôi luôn sẵn sàng lắng nghe ý kiến từ bạn. Hãy điền vào mẫu bên dưới hoặc ghé thăm chúng tôi.' To the left, there is a form for users to fill out their information:

- Họ tên: Nhập họ tên của bạn
- Email: Nhập địa chỉ email
- Nội dung: Bạn muốn nhắn điều gì?

Below the form is a large blue button labeled 'Gửi' (Send).

To the right of the form is a dark blue sidebar containing contact information:

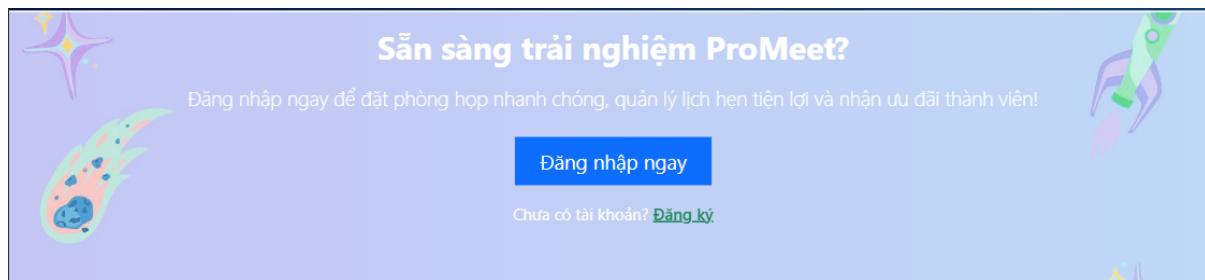
- Địa chỉ: Dĩ An, Bình Dương, Việt Nam
- Email: contact@promeet.vn
- Hotline: + 84 123 456 78
- Giờ làm việc: Thứ 2 – Thứ 7, 8:00 – 17:30

At the bottom right is a Google Map showing the location of the university, with a red pin marking 'Trường Đại học Bách khoa - Đại học...' and a button labeled 'Xem bản đồ lớn hơn' (View larger map).

Hình 21: Phần liên hệ trên trang chủ phía người dùng khách



Phần cuối cùng của trang chủ là một call-to-action (CTA) khuyến khích người dùng đăng nhập hoặc đăng ký tài khoản để có thể sử dụng các tính năng của ProMeet, chẳng hạn như đặt phòng họp và quản lý lịch hẹn.

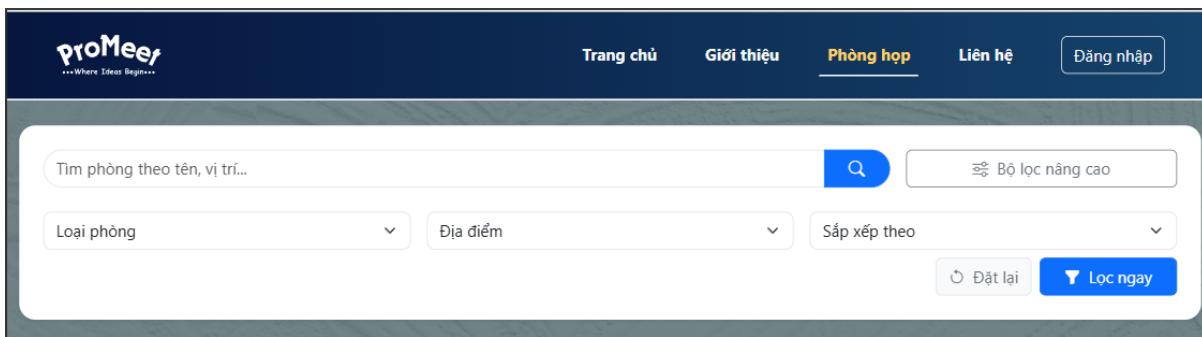


Hình 22: Phần call-to-action khuyến khích đăng nhập trên trang chủ người dùng khách

Người dùng có thể truy cập trang danh sách phòng họp bằng cách bấm “Xem tất cả phòng” tại phần Phòng họp nổi bật. Tại đây, họ có thể tìm kiếm, áp dụng bộ lọc nâng cao và thêm phòng vào danh sách yêu thích qua biểu tượng trái tim.

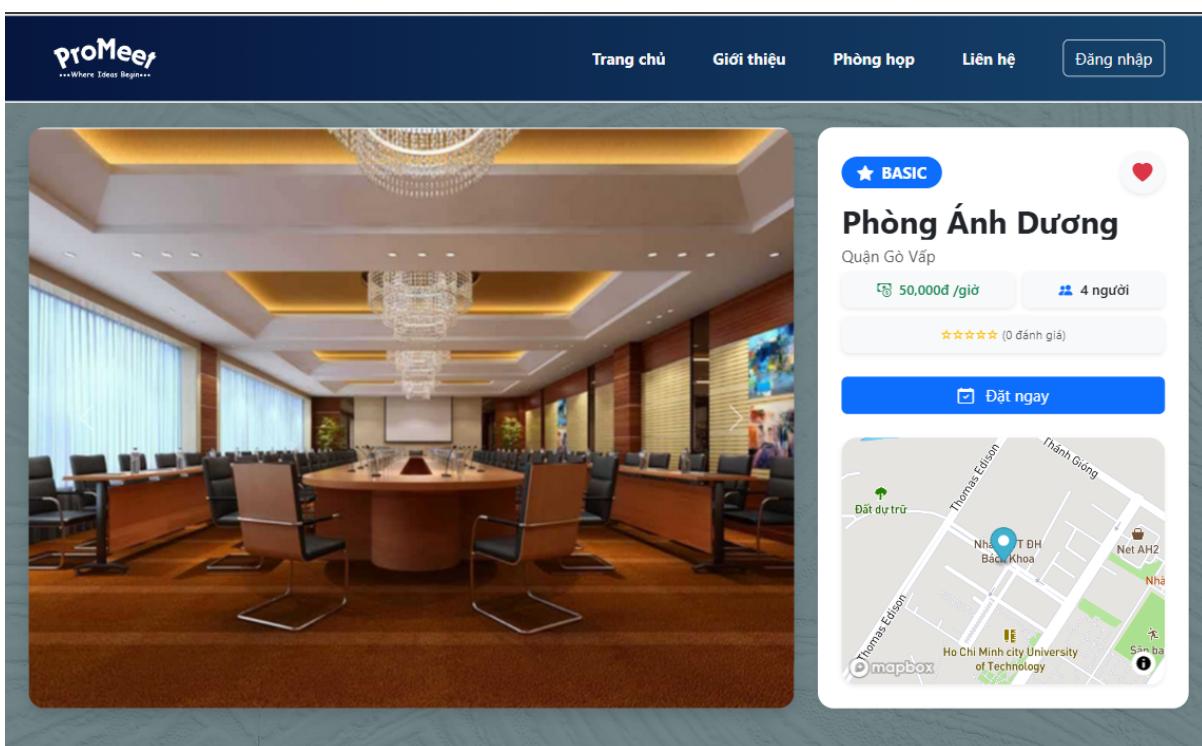
Tên Phòng	Loại Phòng	Mô Tả	Đặt Ngay	Yêu Thích
Phòng Thành Công	Premium	Chưa có đánh giá Quận 1	Đặt ngay	
Phòng Ánh Dương	Basic	Chưa có đánh giá Quận Gò Vấp	Đặt ngay	
Phòng Hoa Mặt Trời	Standard	Chưa có đánh giá Quận Phú Nhuận	Đặt ngay	
Phòng Hoa Anh Đào	Standard	Chưa có đánh giá Quận 3	Đặt ngay	

Hình 23: Giao diện trang danh sách phòng họp



Hình 24: Bộ lọc nâng cao trên trang danh sách phòng

Khi bấm “Đặt ngay” ở một phòng bất kỳ, hệ thống sẽ chuyển đến trang chi tiết phòng, gồm: hình ảnh, thông tin cơ bản, mô tả chi tiết, form đặt phòng, đánh giá người dùng và gợi ý phòng liên quan.



Hình 25: Giao diện thông tin chi tiết phòng

### Mô tả chi tiết

Phòng Ánh Dương là một không gian làm việc lý tưởng với ánh sáng tự nhiên, trang bị đầy đủ tiện nghi như máy chiếu, bảng trắng và Wi-Fi miễn phí. Phòng phù hợp cho các cuộc họp nhỏ và trao đổi công việc hiệu quả.

#### Tiện nghi:

- Máy chiếu
- Bảng trắng
- Wi-Fi miễn phí
- Điều hòa không khí

Hình 26: Giao diện phần mô tả chi tiết của trang chi tiết phòng



**Đặt phòng**

Chọn ngày  
05/08/2025

Chọn khung giờ

08:00 08:30 09:00 09:30 10:00 10:30 11:00 11:30 **12:00** 12:30 13:00 13:30 14:00 14:30 15:00  
15:30 16:00 16:30 17:00 17:30

\* Các khung giờ đã đặt sẽ bị vô hiệu hóa.

Tổng cộng: **0đ**

**⚠ Lưu ý:**

- Thời gian đặt chia theo khung 30 phút.
- Có 30 phút dọn phòng giữa các lượt đặt.
- Bạn chỉ có 10 phút để hoàn tất thanh toán sau khi đặt.
- Các khung giờ đã đặt sẽ bị ẩn hoặc vô hiệu hóa.

**ⓘ Chính sách hủy & hoàn tiền:**

- Hoàn tiền 100% nếu hủy trước 2 giờ so với thời gian bắt đầu.
- Hoàn tiền 50% nếu hủy trong vòng 30 phút **đến** 2 giờ trước thời gian bắt đầu.
- Không hoàn tiền** nếu hủy sau thời điểm trên.
- Thời gian bắt đầu tính theo block đầu tiên bạn đã đặt.
- Hoàn tiền sẽ được xử lý trong vòng 24 giờ sau khi hủy thành công.

Tôi đã đọc và đồng ý với chính sách hủy & hoàn tiền.

**Đặt phòng ngay**

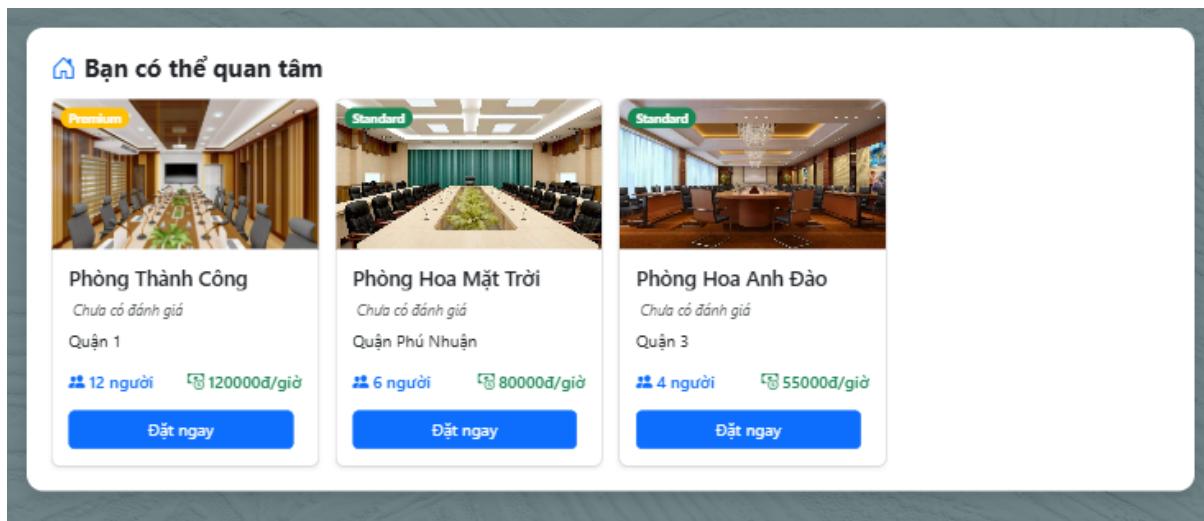
Hình 27: Giao diện phần đặt phòng của trang chi tiết phòng

**★ Trải nghiệm người dùng**

Mới nhất

Chưa có ai kịp đánh giá, bạn có muốn là người tiên phong không? 😊

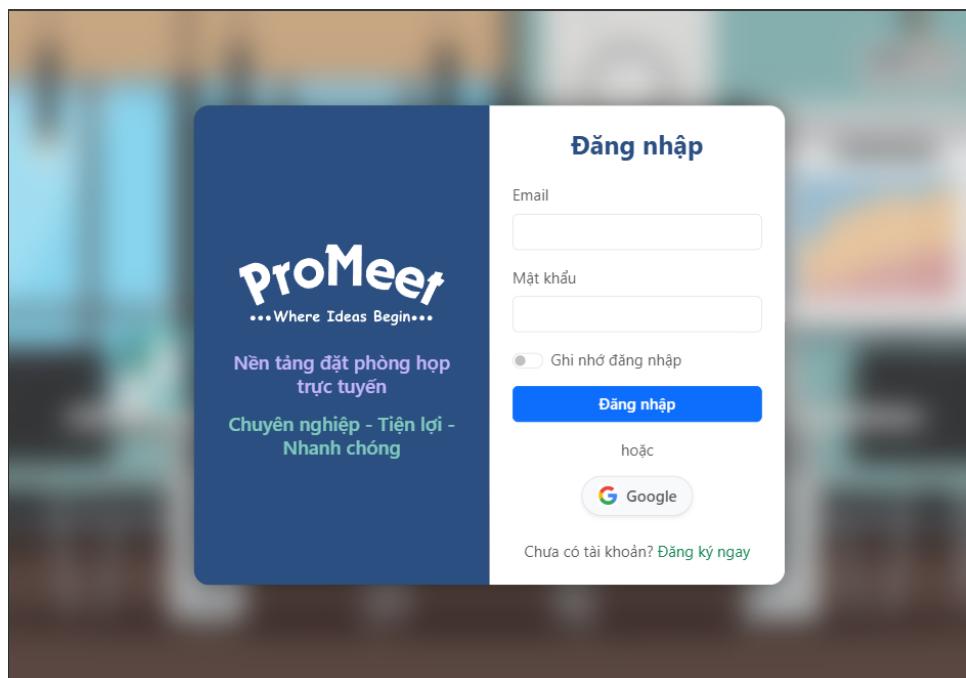
Hình 28: Giao diện phần đánh giá của trang chi tiết phòng



Hình 29: Giao diện phần các phòng liên quan của trang chi tiết phòng

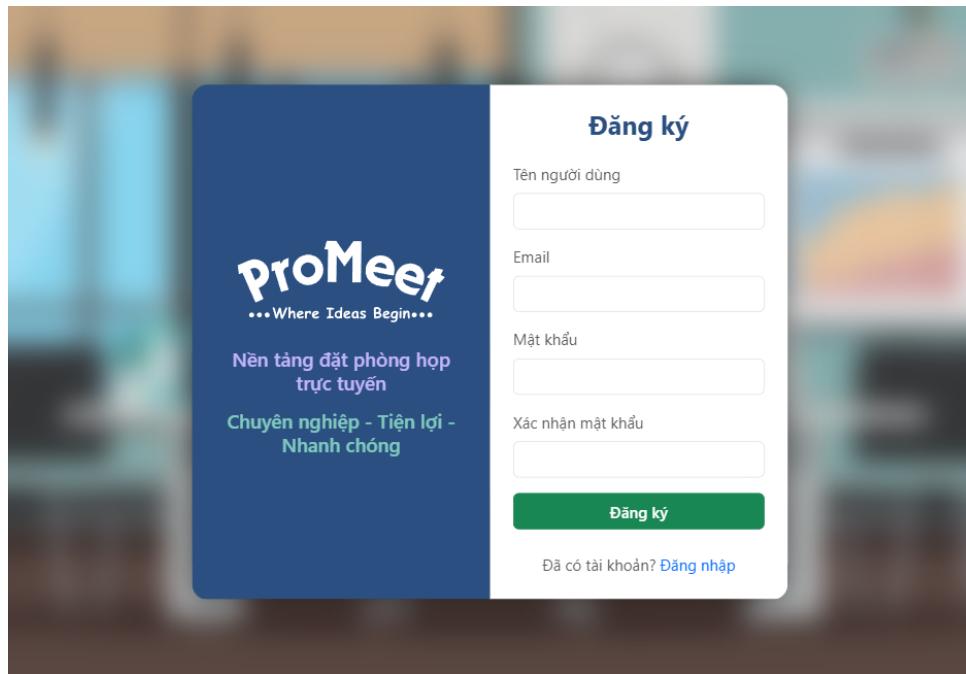
Khách chưa đăng nhập không thể đặt phòng – nếu bấm, sẽ được chuyển đến trang đăng nhập. Người dùng có thể đăng ký tài khoản mới nếu chưa có.

Dưới đây là giao diện của trang đăng nhập



Hình 30: Chức năng đăng nhập

Nếu người dùng chưa có tài khoản, bấm "Đăng kí ngay". Trang đăng kí sẽ được hiện lên:



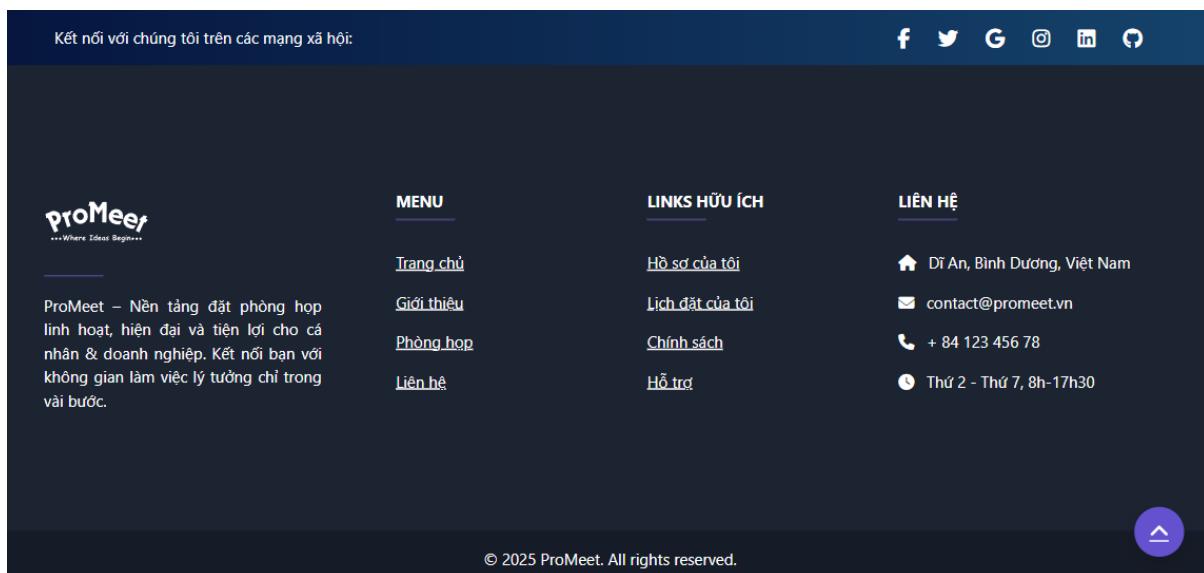
Hình 31: Chức năng đăng ký

## 4.2 Về phía người dùng (đã đăng nhập)

Giao diện người dùng sau khi đăng nhập vẫn giữ nguyên các trang như khách, nhưng có navbar và footer riêng phù hợp với tài khoản đã đăng nhập.



Hình 32: Thanh navbar phía người dùng



Hình 33: Phần footer phía người dùng

Sau khi đăng nhập, người dùng có thể đặt phòng, theo dõi tiến trình thanh toán theo 3 bước: xác



nhận, nhập thông tin liên hệ và thanh toán.

Thanh toán đặt phòng

09:57

1 Xác nhận thông tin 2 Thông tin liên hệ 3 Thanh toán

① Xác nhận thông tin đặt phòng

Phòng: Phòng Ánh Dương  
Địa điểm: Quận Gò Vấp  
Thời gian:

- 08/05/2025 – 14:00 - 14:30
- 08/05/2025 – 14:30 - 15:00
- 08/05/2025 – 15:00 - 15:30
- 08/05/2025 – 15:30 - 16:00

Tổng tiền: **100.000đ**

→ Tiếp tục

Hình 34: Bước 1: Xác nhận thông tin

Thanh toán đặt phòng

09:37

1 Xác nhận thông tin 2 Thông tin liên hệ 3 Thanh toán

② Thông tin liên hệ

Họ và tên  
Nhập họ tên

Email  
Nhập email

Số điện thoại  
Nhập số điện thoại

← Quay lại → Tiếp tục

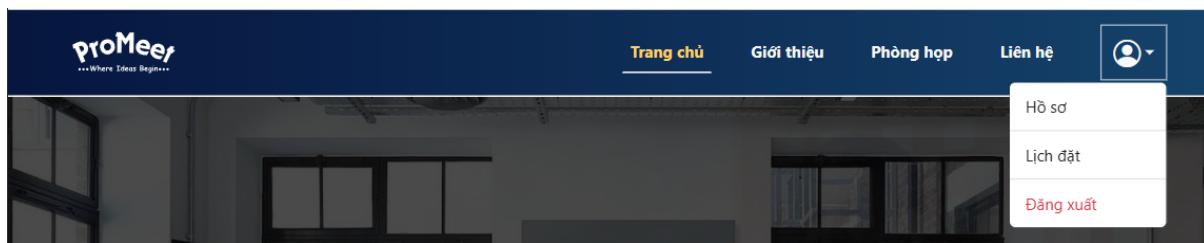
Hình 35: Bước 2: Diền danh sách liên hệ



The screenshot shows a mobile application interface for booking a room. At the top, there is a header with the university logo and the text "TRƯỜNG ĐẠI HỌC BÁCH KHOA ĐHQG-HCM" and "Khoa Khoa học và Kỹ thuật máy tính". Below the header, the title "Thanh toán đặt phòng" is displayed with a checkmark icon. A progress bar at the top indicates three steps: "Xác nhận thông tin" (Step 1), "Thông tin liên hệ" (Step 2), and "Thanh toán" (Step 3). The current step is Step 2. In the main content area, there is a section titled "Chọn phương thức thanh toán" with two options: "Chuyển khoản ngân hàng" (selected) and "Ví MoMo". Below this, there is a QR code for scanning to make the payment. Two buttons are present: "Quay lại" (Back) and "Xác nhận thanh toán" (Confirm payment).

Hình 36: Bước 3: Thanh toán

Ngoài ra, họ có thẻ quản lý tài khoản: chỉnh sửa thông tin cá nhân, đổi mật khẩu, xem lịch sử đặt phòng, đánh giá phòng đã sử dụng và xem lại đánh giá đã viết.



Hình 37: Dropdown user



The screenshot shows a user profile page for Bob Smith. At the top, there is a navigation bar with links for Trang chủ (Home), Giới thiệu (About), Phòng họp (Meeting Room), Liên hệ (Contact), and a user icon. Below the navigation bar is a large circular profile picture placeholder with the ProMeer logo. To the right of the placeholder, the section title "Thông tin cá nhân" (Personal Information) is displayed with a person icon. Under this title, there are four input fields: Họ tên (Name) containing "Bob Smith", Ngày sinh (Date of Birth) containing "05/15/1992", Giới tính (Gender) showing "Nam" (Male), and Email containing "bob@example.com". Below these fields is a "Số điện thoại" (Phone Number) field with the value "0987654321". A blue button labeled "Chỉnh sửa hồ sơ" (Edit Profile) is located below the phone number field. Below this section, the user's name "Bob Smith" is displayed in bold, followed by the text "Thành viên từ 07-05-2025". At the bottom left of the page, there is a link labeled "Đổi mật khẩu" (Change Password) with a lock icon.

Hình 38: Giao diện quản lý thông tin cá nhân

The screenshot shows a password change form titled "Đổi mật khẩu" (Change Password). It has two main sections: "Mật khẩu hiện tại" (Current Password) and "Mật khẩu mới" (New Password). In the "Mật khẩu hiện tại" section, there is a placeholder "Nhập mật khẩu hiện tại". In the "Mật khẩu mới" section, there are two fields: one for "Nhập mật khẩu mới" and another for "Nhập lại mật khẩu mới". A blue button labeled "Cập nhật mật khẩu" (Update Password) is located at the bottom right of the form.

Hình 39: Giao diện khi bấm vào Đổi mật khẩu



The screenshot shows a booking management interface for 'Lịch đặt của bạn – ProMeet'. At the top, there's a navigation bar with links for 'Trang chủ', 'Giới thiệu', 'Phòng họp', 'Liên hệ', and a user profile icon. Below the header is a search bar with placeholder text 'Tim theo mã đặt phòng, tên phòng hoặc ngày...' and a 'Tim kiếm' button. The main content area displays a table of booked rooms:

#	Mã đặt phòng	Tên phòng	Thời gian	Trạng thái	Tổng tiền	Hành động
1	PR202505080001	Phòng Ánh Dương	2025-05-08 – 14:00-14:30	Chờ xác nhận	100,000 đ	<a href="#">Chi tiết</a>

At the bottom of the page, there's a footer bar with social media icons for Facebook, Twitter, Google+, LinkedIn, and Pinterest, along with the text 'Kết nối với chúng tôi trên các mạng xã hội:'.

Hình 40: Giao diện quản lý lịch đặt phòng đã đặt



The screenshot shows a booking management system for 'ProMeet'. At the top, there's a navigation bar with links for 'Trang chủ', 'Giới thiệu', 'Phòng họp', 'Liên hệ', and a user profile icon. Below the header, there's a 'Trở về' (Back) button and a title 'Chi tiết lịch đặt phòng – ProMeet'. A progress bar indicates the booking status: step 1 (green), step 2 (blue), step 3 (yellow), and step 4 (orange). Below the progress bar, it says 'Đặt phòng lúc 12:05 – 08/05/2025' and 'Đã thanh toán lúc 12:08 – 08/05/2025'. A note about cancellation and refund policies follows:

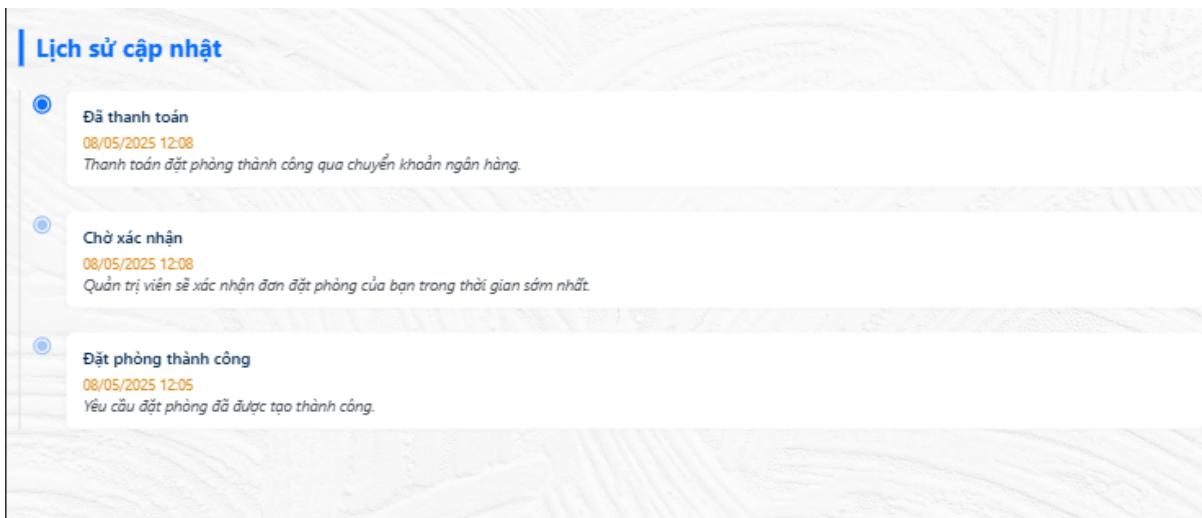
**Chính sách hủy & hoàn tiền:**

- Hoàn tiền 100% nếu hủy trước 2 giờ so với thời gian bắt đầu.
- Hoàn tiền 50% nếu hủy trong vòng 30 phút **đến** 2 giờ trước thời gian bắt đầu.
- Không hoàn tiền** nếu hủy sau thời điểm trên.
- Thời gian bắt đầu tính theo **block đầu tiên** bạn đã đặt.
- Hoàn tiền sẽ được xử lý trong vòng 24 giờ sau khi hủy thành công.

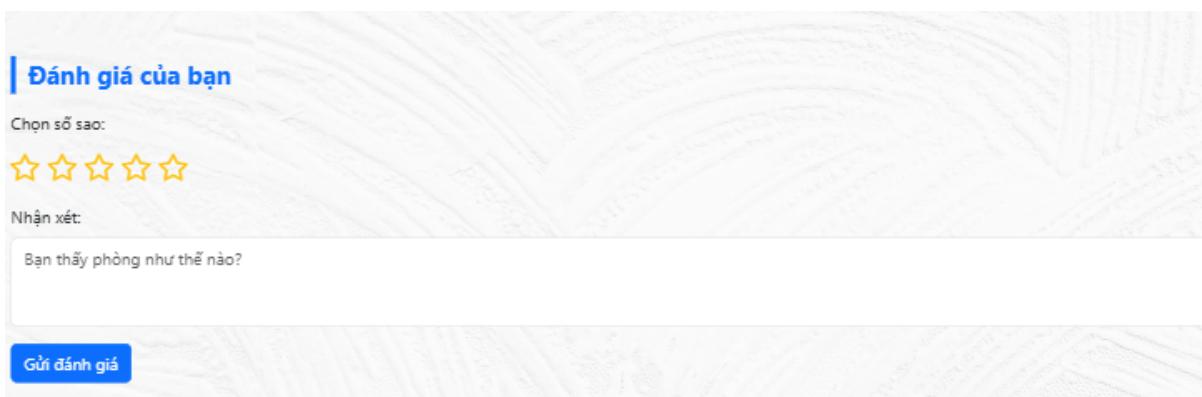
**Thông tin đặt phòng**

<b>Thông tin chung</b>	<b>Mã đặt phòng:</b> PR202505080001 <b>Phòng:</b> Phòng Ánh Dương
<b>Địa điểm:</b> Quận Gò Vấp	<b>Trạng thái:</b> Chờ xác nhận
<b>Thời gian:</b>	<b>Tổng tiền:</b> 100.000đ
• 2025-05-08 - 14:00:00	
• 2025-05-08 - 14:30:00	
• 2025-05-08 - 15:00:00	
• 2025-05-08 - 15:30:00	
<b>Phương thức thanh toán:</b> 0	
<b>Thông tin liên hệ</b>	
<b>Người đặt:</b> Phò	<b>Email:</b> bob@example.com
<b>SĐT:</b> 1234556	

Hình 41: Giao diện quản lý chi tiết 1 đơn đặt lịch



Hình 42: Giao diện quản lý chi tiết 1 đơn đặt lịch (tiếp theo)



Hình 43: Giao diện viết review khi lịch đặt đã hoàn tất



Hình 44: Giao diện xem review khi đã viết review

### 4.3 Về phía quản trị viên

Giao diện quản trị viên của hệ thống được xây dựng dựa trên template Mazer, mang đến một trải nghiệm người dùng hiện đại và dễ sử dụng. Dưới đây là mô tả chi tiết về các tính năng quan trọng trong giao diện quản trị viên:

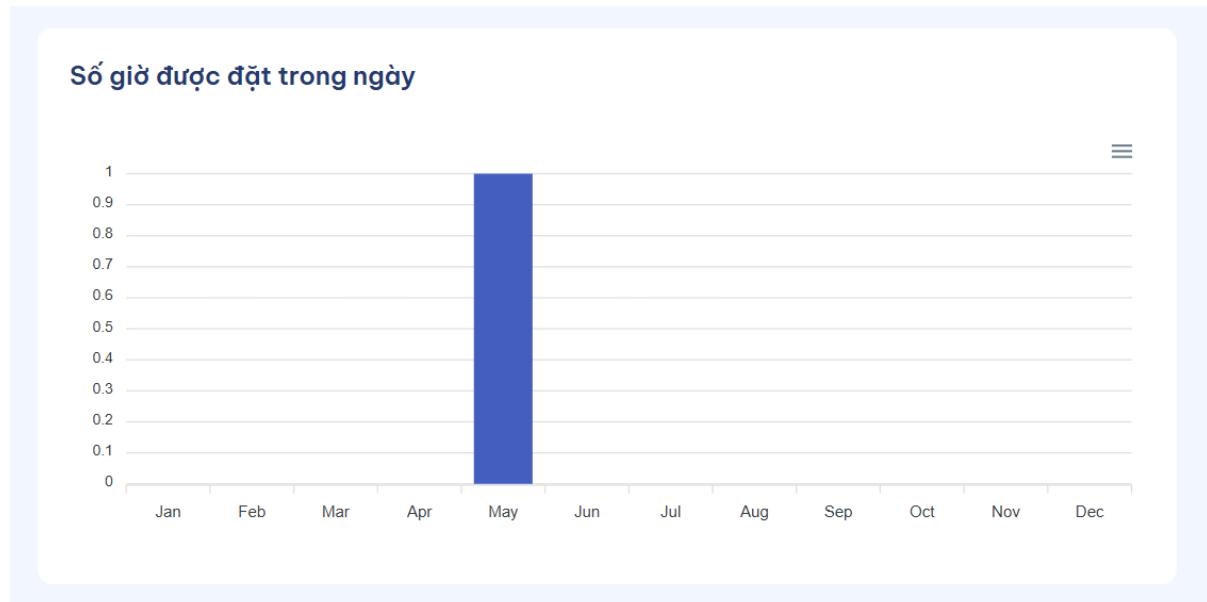


The screenshot shows the Admin Control Panel interface. On the left, there's a sidebar with navigation links: Main, Phòng & Lịch đặt (highlighted), Phòng họp, Lịch đặt, Quản lý người dùng, Khách hàng, Tài khoản, Hồ sơ, Bảo mật, and Đăng xuất. The main content area has a title 'Chào mừng Admin'. It features several cards: 'Người dùng' (2), 'Tổng số phòng' (5), 'Tổng lịch đặt' (6), 'Tổng đánh giá' (2), and a card for 'Quản trị viên @Admin User'. Below these are sections for 'Top 5 phòng nổi bật' (listing Phòng A01, Phòng A02, and Phòng A05) and 'Lịch đặt cần xác nhận' (empty). A pie chart titled 'Người dùng' shows 50.0% for both Nam and Nữ. At the bottom, a box displays 'Số giờ được đặt trong ngày'.

Hình 45: Giao diện bảng điều khiển của admin (chế độ sáng)

The screenshot shows the Admin Control Panel interface in dark mode. The layout is identical to the light mode version, with the same sidebar, title, and cards. The dark theme changes the colors of the cards and the overall background.

Hình 46: Giao diện bảng điều khiển của admin (chế độ tối)



Hình 47: Biểu đồ thống kê tổng số giờ được đặt của tất cả phòng trong 1 tháng

**Bảng điều khiển:** Đây là màn hình chính của quản trị viên, nơi hiển thị các thông tin tổng quan như số lượng phòng họp, tổng số đặt lịch,... Admin có thể chuyển giữa chế độ sáng và chế độ tối để tối ưu hóa trải nghiệm người dùng.

**Quản lý phòng** Products & Orders / Rooms

Theo dõi, chỉnh sửa và quản lý danh sách phòng họp trong hệ thống.

STT	Tên phòng	Loại phòng	Giá/giờ	Địa điểm	Trung bình	Trạng thái	Chi tiết
1	Phòng Ánh Dương	Basic	50,000đ	Quận Gò Vấp	0/5	Hoạt động	Xem
2	Phòng Hoa Anh Đào	Standard	55,000đ	Quận 3	0/5	Hoạt động	Xem
3	Phòng Hoa Mặt Trời	Standard	80,000đ	Quận Phú Nhuận	0/5	Hoạt động	Xem
4	Phòng Thành Công	Premium	120,000đ	Quận 1	0/5	Hoạt động	Xem

Hiển thị 10 dòng Tìm kiếm:

+ Thêm phòng

Hiển thị 1 đến 4 của 4 dòng

Trước 1 Sau

Hình 48: Giao diện quản lý phòng họp

**Quản lý phòng họp:** Giao diện này cho phép admin xem và quản lý tất cả các phòng họp có sẵn trên hệ thống.



### Thêm phòng họp mới

Tên phòng	Giá (VNĐ / giờ)	
Nhập tên phòng	Ví dụ: 300000	
Sức chứa	Phân loại	
Tối đa bao nhiêu người?	-- Chọn loại phòng --	
Địa điểm (Quận/Huyện)	Vĩ độ (Latitude)	Kinh độ (Longitude)
-- Chọn địa điểm --	10.12345	106.12345

Hủy Lưu phòng

Hình 49: Giao diện khi thêm phòng họp mới

**Thêm phòng họp mới:** Khi nhấp vào nút "Thêm phòng", hệ thống sẽ hiện 1 modal để admin nhập thông tin chi tiết về phòng họp mới, bao gồm tên phòng, giá thuê, sức chứa, và các thông tin liên quan.

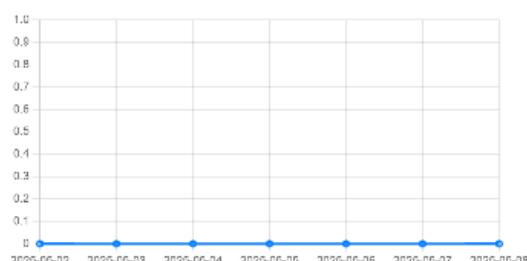


## Quản lý chi tiết phòng

Danh sách phòng / Phòng Ánh Dương

Quản lý thông tin chi tiết của từng phòng.

### Số giờ sử dụng phòng theo ngày



Lượt xem  
6

Tổng lượt đặt  
0

Lượt yêu thích  
1

Điểm trung bình  
0 / 5

Trạng thái



### Thông tin phòng



Tên phòng

Phòng Ánh Dương

Giá (VND / giờ)

50000

Sức chứa

4

Loại

Basic

Địa điểm

Quận Gò Vấp

Vĩ độ

10.881334

Kinh độ

106.806039

Hình 50: Giao diện quản lý chi tiết một phòng họp



Ảnh slideshow

[Thêm ảnh](#)

Mô tả chi tiết

[Lưu](#) [Hủy](#)

File Edit View Insert Format Tools [Upgrade](#)

Phòng Ánh Dương là một không gian làm việc lý tưởng với ánh sáng tự nhiên, trang bị đầy đủ tiện nghi như máy chiếu, bảng trắng và Wi-Fi miễn phí. Phòng phù hợp cho các cuộc họp nhỏ và trao đổi công việc hiệu quả.

Tiện nghi:

- Máy chiếu
- Bảng trắng
- Wi-Fi miễn phí
- Điều hòa không khí

Hình 51: Giao diện quản lý chi tiết phòng họp (tiếp theo)

**Quản lý chi tiết phòng họp:** Giao diện này cho phép admin xem và chỉnh sửa thông tin chi tiết của từng phòng họp. Admin có thể thay đổi các thông tin như giá, sức chứa, loại phòng, ảnh, mô tả chi tiết, thống kê, ẩn/kích hoạt,...

≡

## Quản lý đơn đặt lịch

Rooms & Bookings / Bookings

Quản lý tất cả các đơn đặt phòng, theo dõi trạng thái và lịch sử các đơn đặt.

**Thống kê đơn đặt phòng**

Thời gian: [Toàn bộ](#)

Tình Trạng	Số Lượng
Tổng đơn (*hợp lệ)	6
Chờ thanh toán	9
Chờ xác nhận	1
Đã xác nhận	2
Hoàn tất	1
Đã hủy	2

Hình 52: Giao diện quản lý đặt lịch phòng họp



Danh sách đơn đặt lịchTất cả  mm/dd/yyyy

Hiển thị  dòngTìm kiếm: mã đơn, tên phòng

STT	Mã đặt phòng	Tên phòng	Ngày đặt	Tổng tiền	Trạng thái	Chi tiết
1	PR202505070012	Phòng A01	2025-05-08	75.000đ	<span>Chờ thanh toán</span>	<span>Xem</span>
2	PR202505070011	Phòng A01	2025-05-08	25.000đ	<span>Đã xác nhận</span>	<span>Xem</span>
3	PR202505070010	Phòng A01	2025-05-07	62.500đ	<span>Chờ thanh toán</span>	<span>Xem</span>
4	PR202505070009	Phòng A01	2025-05-08	25.000đ	<span>Chờ thanh toán</span>	<span>Xem</span>
5	PR202505070008	Phòng A01	2025-05-07	25.000đ	<span>Chờ thanh toán</span>	<span>Xem</span>
6	PR202505070007	Phòng A02	2025-05-07	25.000đ	<span>Đã xác nhận</span>	<span>Xem</span>
7	PR202505070006	Phòng A02	2025-05-07	25.000đ	<span>Chờ thanh toán</span>	<span>Xem</span>
8	PR202505070005	Phòng A02	2025-05-07	37.500đ	<span>Chờ thanh toán</span>	<span>Xem</span>
9	PR202505070004	Phòng A02	2025-05-07	37.500đ	<span>Đã hủy</span>	<span>Xem</span>
10	PR202505070003	Phòng A01	2025-05-07	25.000đ	<span>Chờ xác nhận</span>	<span>Xem</span>

Hiển thị 1 đến 10 của 15 dòngTrước 1 2 Sau

Hình 53: Giao diện quản lý các đặt lịch phòng họp tiếp theo

**Quản lý đặt lịch phòng:** Giao diện này cho phép admin quản lý tất cả các yêu cầu đặt phòng từ người dùng cũng như thống kê các trạng thái của các đơn đặt lịch trong hệ thống.



## Chi tiết đơn đặt phòng

Danh sách đơn đặt phòng / #PR202505070011

Quản lý thông tin chi tiết của từng đơn đặt phòng.

### Thông tin đơn hàng

#### THÔNG TIN CHUNG

Mã đặt phòng:	#PR202505070011
Phòng:	Phòng A01
Thời gian:	<ul style="list-style-type: none"><li>08/05/2025 – 11:30 - 12:00</li><li>08/05/2025 – 12:00 - 12:30</li></ul>
Tổng tiền:	25.000đ
Thanh toán:	Ví Momo
Trạng thái:	<span>Đã xác nhận</span>

#### THÔNG TIN LIÊN HỆ

Họ tên:	Trần QUANG tACS
Số điện thoại:	0355999723
Email:	quangtac@example.com

#### TÀI KHOẢN ĐẶT

User ID:	#41201a8e-2a9e-11f0-b0e0-b4450636e7a9
Username:	Bob Smith
Hỗ trợ:	<a href="#">Xem tài khoản</a>

### Timeline xử lý yêu cầu đặt phòng

- Đặt phòng thành công** 07/05/2025 16:14  
*Yêu cầu đặt phòng đã được tạo thành công.*
- Đã thanh toán** 07/05/2025 16:15  
*Thanh toán đặt phòng thành công qua ví MoMo.*
- Chờ xác nhận** 07/05/2025 16:15  
*Quản trị viên sẽ xác nhận đơn đặt phòng của bạn trong thời gian sớm nhất.*
- Đã xác nhận lịch** 07/05/2025 16:25  
*Lịch đặt phòng đã được xác nhận bởi quản trị viên. Cảm ơn quý khách đã tin tưởng ProMeet! Quý khách vui lòng có mặt đúng giờ để sử dụng dịch vụ.*

Hình 54: Giao diện quản lý chi tiết đơn đặt lịch phòng họp

### Hành động

Xác nhận  Hoàn tất  Hủy  Ghi mốc thời gian

Hình 55: Các thao tác cập nhật trạng thái đơn đặt lịch

**Quản lý chi tiết đơn đặt lịch:** Giao diện này cho phép admin xem và quản lý thông tin của từng đơn đặt lịch. Đồng thời cập nhật các trạng thái của đơn đặt lịch (Xác nhận, Hủy, Hoàn tất, Viết thêm timeline)



## Quản lý người dùng

Theo dõi và quản lý danh sách người dùng trong hệ thống.

### Danh sách người dùng

Hiển thị 10 dòng

Tìm kiếm:

STT	Họ và tên	Email	SĐT	Ngày sinh	Giới tính	Trạng thái	Thao tác
1	Bob Smith	bob@example.com	0987654321	15/5/1992	Nam	<span>Hoạt động</span>	<span>Chặn</span>
2	Alice Johnson	alice@example.com	0999888777	20/8/1995	Nữ	<span>Hoạt động</span>	<span>Chặn</span>

Hiển thị 1 đến 2 của 2 dòng

Trước 1 Sau

2025 © ProMeet

Created by TacsQuang

Hình 56: Giao diện quản lý người dùng

**Quản lý người dùng:** Giao diện quản lý người dùng cho phép admin xem thông tin của các người dùng đã đăng ký. Ngoài ra cho phép đặt lại mật khẩu hay cấm 1 người dùng.

## Thông tin cá nhân

Tài khoản / Hồ sơ

Trang chỉnh sửa thông tin cá nhân

**Admin User**  
Quản trị viên ProMeet

**Họ và tên:** Admin User

**Email:** admin@promeet.com

**Số điện thoại:** 0123456789

**Ngày sinh:** 07/28/2004

**Giới tính:** Nam

**Lưu thay đổi**

Hình 57: Giao diện quản lý thông tin cá nhân của admin

**Quản lý thông tin cá nhân của admin:** Admin có thể chỉnh sửa thông tin cá nhân của mình thông qua giao diện này.



## Bảo mật tài khoản

Tài khoản / Bảo mật

Trang quản lý các tùy chọn bảo mật tài khoản

### Đổi mật khẩu

Mật khẩu hiện tại

Nhập mật khẩu hiện tại của bạn

Mật khẩu mới

Nhập mật khẩu mới của bạn

Xác nhận mật khẩu

Nhập mật khẩu xác nhận

Lưu thay đổi

### Xác thực hai bước

Email hiện tại

admin@promeet.com

Lưu thay đổi

Hình 58: Giao diện quản lý bảo mật tài khoản của admin

**Quản lý bảo mật tài khoản:** Giao diện bảo mật tài khoản cho phép admin thay đổi mật khẩu và thay đổi mail.

## 5 Cách cài đặt ứng dụng

Để cài đặt và triển khai ứng dụng ProMeet trong môi trường máy tính cá nhân, bạn cần thực hiện theo các bước sau:

### Bước 1: Cài đặt XAMPP

XAMPP là phần mềm tích hợp Apache, MySQL, PHP và Perl giúp tạo máy chủ web cục bộ. Truy cập trang chủ XAMPP tại <https://www.apachefriends.org/index.html>, chọn phiên bản phù hợp với hệ điều hành và tiến hành cài đặt như phần mềm thông thường.



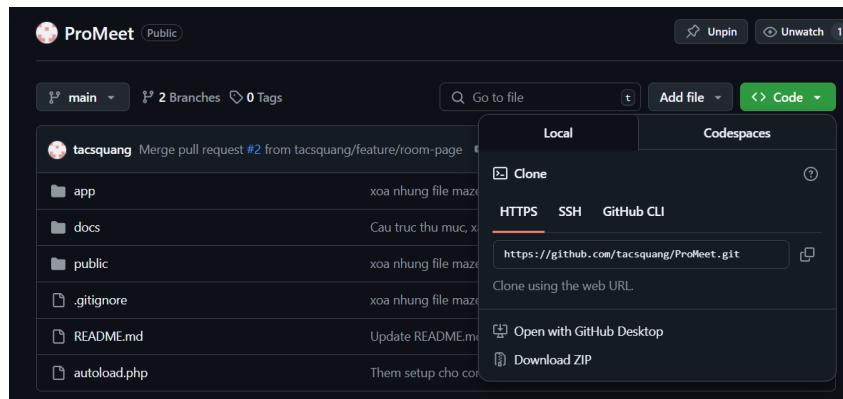
Hình 59: Cài đặt XAMPP

## Bước 2: Clone mã nguồn ProMeet

Sau khi cài đặt XAMPP, bạn cần tải mã nguồn ứng dụng ProMeet từ GitHub. Mở Terminal (hoặc Git Bash trên Windows), di chuyển đến thư mục `htdocs` trong thư mục cài đặt XAMPP, sau đó chạy lệnh:

```
cd C:/xampp/htdocs  
git clone https://github.com/tacsquang/ProMeet.git
```

Sau khi clone thành công, mã nguồn sẽ nằm trong thư mục `htdocs/ProMeet`.

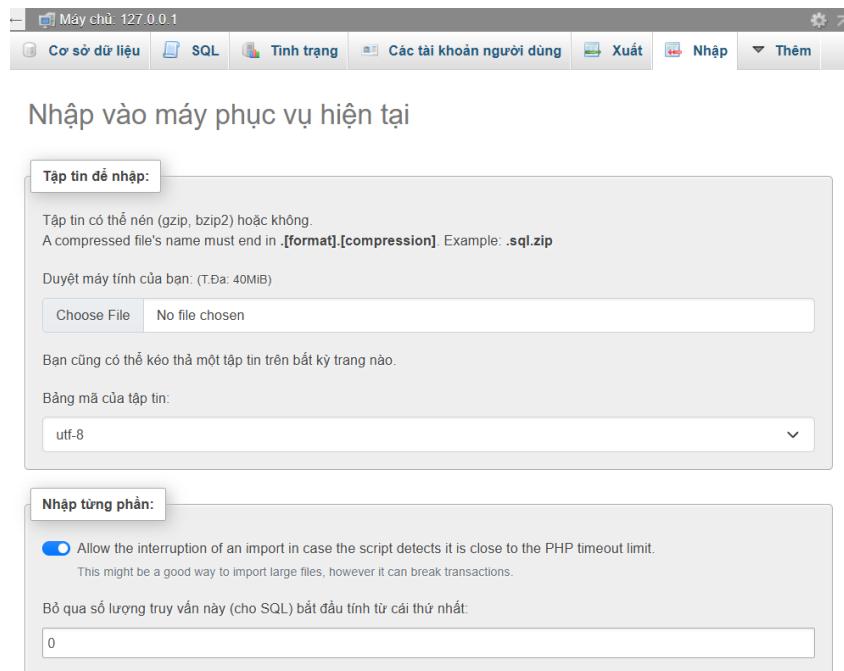


Hình 60: Clone mã nguồn ProMeet vào thư mục `htdocs`

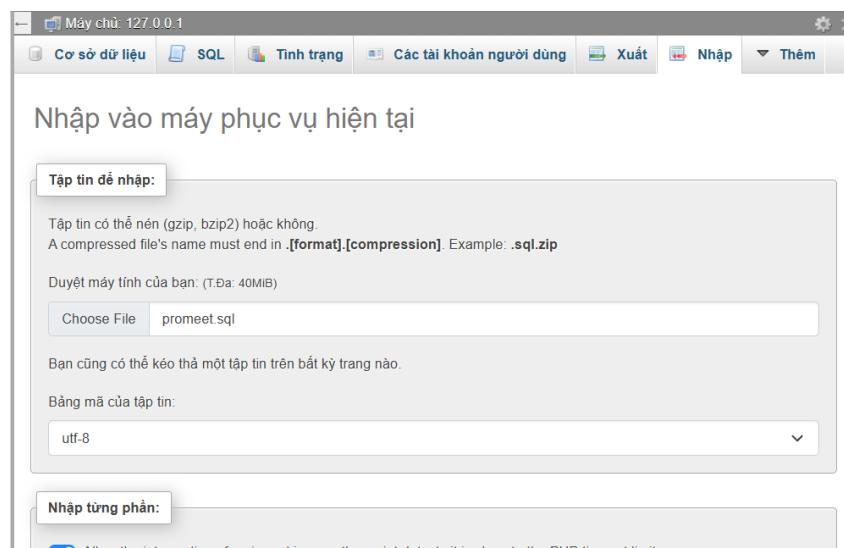
## Bước 3: Tạo cơ sở dữ liệu MySQL

Mở trình duyệt và truy cập <http://localhost/phpmyadmin>, sau đó:

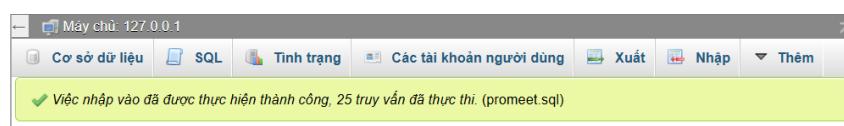
- Nhập vào tab **Import**.
- Chọn file SQL có sẵn trong thư mục mã nguồn (chứa các lệnh `DROP DATABASE`, `CREATE DATABASE`, `CREATE TABLE`).
- Nhấn **Go** để thực hiện import.



Hình 61: Nhập vào import trong phpMyAdmin



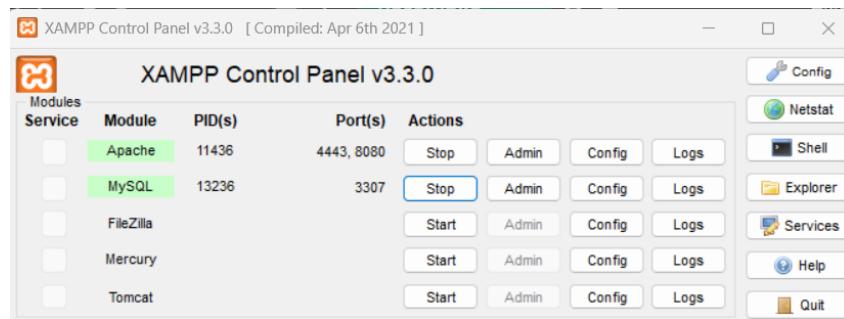
Hình 62: Tải file SQL lên để tạo database



Hình 63: Tạo database thành công

#### Bước 4: Khởi động Apache và MySQL

Mở XAMPP Control Panel, nhấn nút Start cho hai dịch vụ **Apache** và **MySQL**.



Hình 64: Khởi động Apache và MySQL bằng XAMPP

### Bước 5: Cấu hình kết nối cơ sở dữ liệu

Mở file cấu hình trong thư mục ứng dụng tại đường dẫn: config/database.php và kiểm tra hoặc điều chỉnh thông tin như sau:

```
return [  
    'host'      => 'localhost',  
    'port'      => 3306,  
    'database'  => 'prooo_meet',  
    'username'  => 'root',  
    'password'  => '',  
    'charset'   => 'utf8mb4',  
];
```

**Lưu ý:** Trong XAMPP mặc định, tài khoản MySQL là root và không có mật khẩu. Nếu bạn đã đặt mật khẩu MySQL, hãy cập nhật lại phần password tương ứng.

```
config > database.php  
1  <?php  
2  
3  return [  
4  |     'host'      => 'localhost',  
5  |     'port'      => 3306,  
6  |     'database'  => 'pro_meet',  
7  |     'username'  => 'root',  
8  |     'password'  => '',  
9  |     'charset'   => 'utf8mb4',  
10 |];  
11  
12
```

Hình 65: Cấu hình kết nối cơ sở dữ liệu

Sau khi hoàn tất, bạn có thể truy cập ứng dụng tại địa chỉ: <http://localhost/ProMeet/public>

## 6 Nhiệm vụ của từng thành viên

STT	Họ và tên	MSSV	Nhiệm vụ	Phần trăm đóng góp
1	Trần Quang Tác	2212962	FE + BE + Báo cáo	100%



## 7 Source code BTL

Tất cả mã nguồn của dự án, bao gồm frontend, backend, và cơ sở dữ liệu, có thể được truy cập và tải về từ GitHub. Để tham khảo các chi tiết về cách triển khai, vui lòng truy cập vào repository của dự án tại [ProMeet GitHub Repository](#).

### Tài liệu tham khảo

- [1] Bootstrap. Tài liệu Bootstrap. <https://getbootstrap.com/docs/5.3/>.
- [2] SweetAlert2. Tài liệu SweetAlert2. <https://sweetalert2.github.io/>.
- [3] AOS. Tài liệu AOS (Animate On Scroll). <https://michalsnik.github.io/aos/>.
- [4] jQuery. Tài liệu jQuery. <https://api.jquery.com/>.
- [5] TinyMCE. Tài liệu TinyMCE. <https://www.tiny.cloud/docs/>.
- [6] Chart.js. Tài liệu Chart.js. <https://www.chartjs.org/docs/latest/>.
- [7] Simple DataTables. Tài liệu Simple DataTables. <https://simple-datatables.net/>.
- [8] PHP. Tài liệu PHP. <https://www.php.net/manual/en/>.
- [9] MySQL. Tài liệu MySQL. <https://dev.mysql.com/doc/>.
- [10] XAMPP. Tài liệu XAMPP. <https://www.apachefriends.org/index.html>.
- [11] Google. Tài liệu SEO - Hướng dẫn cho các nhà phát triển web. <https://developers.google.com/search/docs/beginner/seo-starter-guide>.
- [12] DVWA. Damn Vulnerable Web Application (DVWA). <http://www.dvwa.co.uk/>.