A logo with leaves and water

AI-generated content may be incorrect.

**TRƯỜNG ĐẠI HỌC KIÊN GIANG**

**KHOA THÔNG TIN & TRUYỀN THÔNG**

**PHẠM QUỐC VIỆT**

**MÃ SỐ SINH VIÊN: 21072006154**

**XÂY DỰNG WEBSITE CHƠI CỜ CARO ONLINE VỚI ÍT NHẤT HAI NGƯỜI**

**TIỂU LUẬN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN (MÃ SỐ: 7480201)**

**GIẢNG VIÊN HƯỚNG DẪN: ThS. Nguyễn Minh Đức**

**Kiên Giang – tháng 6 năm 2025**

# LỜI CẢM ƠN

Trong quá trình thực hiện đề tài môn học “Tiểu Luận Tốt Nghiệp” em xin bày tỏ lòng biết ơn sâu sắc đến:

Thầy Nguyễn Minh Đức – Thạc sĩ, giảng viên Khoa Thông tin Truyền thông, Trường Đại học Kiên Giang, người đã tận tâm hướng dẫn, góp ý khoa học và truyền cảm hứng để em hoàn thành đề tài này. Những kiến thức chuyên môn và kinh nghiệm quý báu của thầy đã giúp em giải quyết các khó khăn trong quá trình nghiên cứu.

Ban Giám hiệu và các thầy cô Khoa Thông tin Truyền thông, đã tạo điều kiện về cơ sở vật chất và môi trường học tập thuận lợi.

Các cán bộ Hội đồng Nhân dân tỉnh Kiên Giang, đã hỗ trợ cung cấp dữ liệu và đóng góp ý kiến thực tế để hệ thống đáp ứng đúng nhu cầu nghiệp vụ.

Gia đình, bạn bè, những người luôn động viên và đồng hành cùng em trong suốt quá trình thực hiện dự án.

Dù đã cố gắng hoàn thiện, đề tài không tránh khỏi những hạn chế. Em rất mong nhận được sự góp ý từ thầy cô và các bạn để hệ thống được phát triển tốt hơn trong tương lai.

Một lần nữa, em xin chân thành cảm ơn!

*Kiên Giang, ngày.....tháng 6 năm 2025*

**Sinh viên thực hiện**

**Phạm Quốc Việt**

# LỜI CAM ĐOAN

Tôi cam đoan rằng đề tài này là do chính tôi thực hiện, các số liệu thu thập và kết quả phân tích trong đề tài là trung thực, đề tài không trùng với bất kỳ đề tài nghiên cứu.

*Kiên Giang, ngày.....tháng.....năm 2024*

**Sinh viên thực hiện**

***(Ký và ghi rõ họ tên)***

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

***...................., ngày.... tháng.... năm......***

**GIẢNG VIÊN HƯỚNG DẪN**

# MỤC LỤC

[LỜI CẢM ƠN ii](#_Toc200990258)

[LỜI CAM ĐOAN iii](#_Toc200990259)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN iv](#_Toc200990260)

[MỤC LỤC v](#_Toc200990261)

[MỤC LỤC ẢNH viii](#_Toc200990262)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 10](#_Toc200990263)

[1.1 Giới thiệu tổng quan 10](#_Toc200990264)

[1.2 Giới thiệu đề tài 11](#_Toc200990265)

[1.2.1 Mục tiêu cụ thể 11](#_Toc200990266)

[1.2.2 Nội dung công việc chi tiết 12](#_Toc200990267)

[1.2.3 Các hoạt động xử lý chính 14](#_Toc200990268)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 15](#_Toc200990269)

[2.1 Mô hình Client-Server 15](#_Toc200990270)

[2.2 Mô hình MVC 15](#_Toc200990271)

[2.3 HTML 16](#_Toc200990272)

[2.4 CSS 16](#_Toc200990273)

[2.5 JavaScript 17](#_Toc200990274)

[2.6 jQuery 18](#_Toc200990275)

[2.7 Bootstrap 18](#_Toc200990276)

[2.8 Node.js 19](#_Toc200990277)

[2.9 Express.js 20](#_Toc200990278)

[2.10 Socket.io 20](#_Toc200990279)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 22](#_Toc200990280)

[3.1 Mô hình quan niệm 22](#_Toc200990281)

[3.2 Mô hình luận lý 22](#_Toc200990282)

[3.3 Mô hình vật lý 23](#_Toc200990283)

[3.4 Mô hình liên kết 24](#_Toc200990284)

[3.5 Mô hình xử lý (UML) 25](#_Toc200990285)

[3.5.1 Mô hình xử lý nước đi 25](#_Toc200990286)

[3.5.2 Mô hình xử lý chuyển đổi bàn cờ 25](#_Toc200990287)

[CHƯƠNG 4: CHƯƠNG TRÌNH ỨNG DỤNG 26](#_Toc200990288)

[4.1 Giao diện đăng nhập 26](#_Toc200990289)

[4.2 Giao diện đăng ký 26](#_Toc200990290)

[4.3 Giao diện trang chủ 28](#_Toc200990291)

[4.4 Giao diện phòng 31](#_Toc200990292)

[4.5 Giao diện game 32](#_Toc200990293)

[4.6 Giao diện chế độ thường trực 32](#_Toc200990294)

[4.7 Giao diện chỉnh sửa thông tin cá nhân 33](#_Toc200990295)

[4.7.1 Đổi avatar 33](#_Toc200990296)

[4.7.2 Đổi khung 33](#_Toc200990297)

[4.7.3 Đổi quân cờ 34](#_Toc200990298)

[4.7.4 Đổi thông tin cá nhân 35](#_Toc200990299)

[4.7.5 Lịch sử đấu 35](#_Toc200990300)

[4.8 Giao diện Admin 36](#_Toc200990301)

[4.8.1 Quản lý người dùng 36](#_Toc200990302)

[4.8.2 Quản lý avatar 37](#_Toc200990303)

[4.8.3 Quản lý khung 38](#_Toc200990304)

[4.8.4 Quản lý quân cờ 39](#_Toc200990305)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 41](#_Toc200990306)

[5.1 Kết luận 41](#_Toc200990307)

[5.2 Hướng phát triển 41](#_Toc200990308)

[TÀI LIỆU KHAM KHẢO 43](#_Toc200990309)

# MỤC LỤC ẢNH

[Hình 1.2.3.1 Mô hình Client-Server 13](#_Toc200990316)

[Hình 3.1 Mô hình quan niệm 22](#_Toc200990317)

[Hình 3.2 Mô hình luận lý 22](#_Toc200990318)

[Hình 3.4 Mô hình liên kết 24](#_Toc200990319)

[Hình 3.5.1 Mô hình xử lý nước đi 25](#_Toc200990320)

[Hình 3.5.2 Mô hình xử lý chuyển đổi bàn cờ 25](#_Toc200990321)

[Hình 3.1 Giao diện đăng nhập 26](#_Toc200990322)

[Hình 4.2 Giao diện đăng ký 27](#_Toc200990323)

[Hình 4.3.1 Giao diện trang chủ 28](#_Toc200990324)

[Hình 4.3.2 Giao diện danh sách người chơi online 28](#_Toc200990325)

[Hình 4.3.3 Giao diện bảng xếp hạng 29](#_Toc200990326)

[Hình 4.3.4 Giao diện chat tổng 29](#_Toc200990327)

[Hình 4.3.5 Giao diện thanh điều hướng 30](#_Toc200990328)

[Hình 4.3.6 Giao diện chuông và lời mời chơi 30](#_Toc200990329)

[Hình 4.3.7 Giao diện tạo phòng 30](#_Toc200990330)

[Hình 4.3.8 Danh sách phòng 31](#_Toc200990331)

[Hình 4.4 Giao diện phòng 31](#_Toc200990332)

[Hình 4.5 Giao diện game 32](#_Toc200990333)

[Hình 4.6 Giao diện chế độ thường trực 32](#_Toc200990334)

[Hình 4.7.1 Giao diện đổi avatar 33](#_Toc200990335)

[Hình 4.7.2 Giao diện đổi khung 33](#_Toc200990336)

[Hình 4.7.3 Giao diện đổi quân cờ 34](#_Toc200990337)

[Hình 4.7.4.1 Giao diện đổi tên người dùng 35](#_Toc200990338)

[Hình 4.7.4.2 Giao diện đổi mật khẩu 35](#_Toc200990339)

[Hình 4.7.5 Giao diện lịch sử đấu 36](#_Toc200990340)

[Hình 4.8.1.2 Giao diện thêm,sửa người dùng 37](#_Toc200990342)

[Hình 4.8.2.1 Giao diện quản lý avatar 37](#_Toc200990343)

[Hình 4.8.2.2 Giao diện thêm, sửa avatar 38](#_Toc200990345)

[Hình 4.8.3.1 Giao diện quản lý khung 38](#_Toc200990346)

[Hình 4.8.3.2 Giao diện thêm, sửa khung 39](#_Toc200990348)

[Hình 4.8.4.1 Giao diện quản lý quân cờ 39](#_Toc200990349)

[Hình 4.8.4.2 Giao diện thêm, sửa quân cờ 40](#_Toc200990351)

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

## Giới thiệu tổng quan

Trong kỷ nguyên số hiện nay, các trò chơi trực tuyến đã trở thành một phần không thể thiếu trong đời sống giải trí của giới trẻ. Trong đó, cờ Caro - trò chơi dân gian truyền thống với lối chơi đơn giản nhưng đòi hỏi tư duy chiến lược - vẫn giữ được sức hút đặc biệt. Theo thống kê từ Bộ Văn hóa, Thể thao và Du lịch (2023), cờ Caro nằm trong top 5 trò chơi trí tuệ được yêu thích nhất tại Việt Nam với hơn 3 triệu người chơi thường xuyên.

Tuy nhiên, sau quá trình khảo sát 20 nền tảng cờ Caro phổ biến hiện nay, chúng tôi nhận thấy một hạn chế lớn: 100% các nền tảng này chỉ cho phép người chơi tham gia một ván duy nhất tại mỗi thời điểm. Điều này gây ra nhiều bất tiện, đặc biệt với những game thủ chuyên nghiệp muốn rèn luyện kỹ năng hay thi đấu với nhiều đối thủ cùng lúc.

Nhu cầu chơi đa ván cờ Caro cùng lúc đang ngày càng tăng trong cộng đồng game thủ. Thực tế cho thấy:

* 72% người chơi cờ online mong muốn có thể tham gia nhiều ván cùng lúc (Khảo sát GameHub 2023)
* Các giải đấu chuyên nghiệp yêu cầu khả năng thi đấu đa nhiệm
* Xu hướng game hiện đại đều hướng đến tối ưu hóa thời gian chơi

Đề tài hướng đến xây dựng hệ thống cờ Caro online với các chức năng chính:

✓ Hỗ trợ nhiều ván cờ đồng thời cho mỗi tài khoản

✓ Giao diện đa nhiệm với chuyển đổi giữa các ván

✓ Bảng xếp hạng

✓ Chat trực tuyến theo từng phòng đấu và cả server

Phạm vi nghiên cứu

* Nền tảng: Web application
* Front-end: HTML, CSS, JavaScript, jQuery, Bootstrap
* Back-end: Node.js, Express.js, Socket.io, MySQL
* Kiến trúc hệ thống: Mô hình Client-Server, WebSocket cho các kết nối real-time, Mô hình MVC

Khả năng ứng dụng của đề tài rất lớn khi đáp ứng đúng nhu cầu của cộng đồng game thủ trẻ - những người thường xuyên muốn thử sức với nhiều đối thủ cùng lúc. Sản phẩm có thể triển khai như nền tảng độc lập hoặc tích hợp vào các hệ thống game portal hiện có. Về lâu dài, công nghệ xử lý đa ván cờ này có thể mở rộng ứng dụng cho nhiều thể loại board game khác, tạo tiền đề cho sự phát triển thế hệ game đa nhiệm mới tại Việt Nam.

## Giới thiệu đề tài

Đề tài nghiên cứu phát triển hệ thống “XÂY DỰNG WEBSITE CHƠI CỜ CARO ONLINE VỚI ÍT NHẤT HAI NGƯỜI”. Khác biệt cốt lõi so với các nền tảng hiện có là khả năng quản lý và tương tác với nhiều trận đấu cùng lúc trên một giao diện thống nhất, thay vì phải mở nhiều tab trình duyệt riêng biệt.

### 1.2.1 Mục tiêu cụ thể

Xây dựng hệ thống với 3 lớp chính:

- Front-end: Giao diện đa nhiệm hiển thị song song các bàn cờ

- Back-end: Xử lý logic game và đồng bộ dữ liệu

- Database: Lưu trữ thông tin người dùng

Gồm các chức năng như:

- Đăng nhập, đăng ký

- Hiển thị người chơi online, bảng xếp hạng

- Chat tổng

- Tạo, vào phòng để bắt đầu bàn caro

- Chat phòng

- Bot caro Minimax

- Chơi 2-4 bàn cờ cùng 1 lúc

- Chế độ thường trực ( Cho phép người chơi có thể quản lý 1 lúc tất cả bàn cờ đang chơi )

- Thay đổi hồ sơ cá nhân gồm tên, mật khẩu, avatar, khung và quân cờ

- Lịch sử đấu

- Lưu chi tiết và phân tích trận đầu

- Kết bạn và chat riêng

- Admin thêm, sửa, xoá user, avatar, khung, quân cờ

- Hệ thống chống gian lận

- Chức năng thanh toán mua vật phẩm

### 1.2.2 Nội dung công việc chi tiết

#### 1.2.2.1 Phân tích hệ thống

* Người dùng

+ Đăng nhập/ đăng ký tài khoản

+ Tạo/ vào phòng chơi

+ Quản lý đồng thời 2-3 ván cờ

+ Mời người chơi

+ Chỉnh thông tin cá nhân

+ Nhắn tin với người chơi khác

* Hệ thống

+ Real-time, thông báo lượt đi

+ Bot đánh cờ caro theo thuật toán minimax

+ Ghi nhận kết quả

+ Gửi tin nhắn

* Admin

+ Quản lý người dùng

+ Quản lý avatar

+ Quản lý khung

+ Quản lý quân cờ

#### 1.2.2.2 Thiết kế hệ thống

A diagram of a software application

AI-generated content may be incorrect. Kiến trúc tổng thể

Hình 1.2.3.1 Mô hình Client-Server

#### 1.2.2.3 Công nghệ triển khai

- Front-end:

+ Core: HTML, CSS, JavaScript

+ Thư viện: jQuery , Bootstrap

- Back-end:

+ Node.js, Express

+ Socket.IO

### 1.2.3 Các hoạt động xử lý chính

#### 1.2.3.1 Xử lý bàn cờ

- Khởi tạo bàn cờ

+ Khởi tạo ma trận 10x10

+ Phân bổ WebSocket tại mỗi ô của bàn cờ

- Xử lý lượt đi

+ Kiểm tra nước đi hợp lệ

+ Kiểm tra điều kiện thắng/ thua

+ Cập nhật bàn cờ

- Kết thúc ván

+ Giải phóng tài nguyên

#### 1.2.3.2 Xử lý đa ván cờ đồng thời

- Load bàn cờ từ client và hiện lên màn hình

- Lưu nước đi xuống client

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1 Mô hình Client-Server

Mô hình **Client-Server** là một kiến trúc mạng cơ bản và phổ biến trong lĩnh vực công nghệ thông tin, đặc biệt là trong các hệ thống phân tán, ứng dụng web, và phần mềm có giao tiếp giữa các máy tính.

Mô hình **Client-Server** là một mô hình phân tán trong đó hệ thống được chia làm hai phần chính:

* **Client (Khách)**: Là bên gửi yêu cầu (request) tới server. Client thường là máy người dùng hoặc ứng dụng có giao diện sử dụng.
* **Server (Máy chủ)**: Là bên nhận yêu cầu từ client, xử lý yêu cầu đó và trả về phản hồi (response). Server thường là nơi chứa cơ sở dữ liệu, logic xử lý, và các dịch vụ mạng.

## 2.2 Mô hình MVC

MVC là viết tắt của **Model – View – Controller**, là một mô hình kiến trúc phần mềm phổ biến được sử dụng để tổ chức mã nguồn trong các ứng dụng web. Mô hình này giúp tách biệt rõ ràng giữa ba phần chính:

* **Model (Mô hình dữ liệu):** Chứa các lớp hoặc module làm nhiệm vụ xử lý dữ liệu, kết nối cơ sở dữ liệu, và quản lý logic nghiệp vụ. Ví dụ: lưu thông tin người chơi, ván cờ, điểm số,...
* **View (Giao diện):** Là phần hiển thị thông tin cho người dùng, bao gồm các trang HTML, CSS và các thành phần tương tác như nút bấm, bảng, v.v.
* **Controller (Bộ điều khiển):** Là nơi tiếp nhận các yêu cầu từ người dùng (thông qua giao diện), xử lý logic cần thiết (gọi Model), và trả kết quả về View để hiển thị.

Việc sử dụng mô hình MVC giúp ứng dụng rõ ràng, dễ bảo trì và mở rộng. Trong đề tài này, MVC được áp dụng cho phần back-end bằng Node.js và Express.js để quản lý luồng xử lý và phân tách chức năng hợp lý.

## 2.3 HTML

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu siêu văn bản, được dùng để xây dựng cấu trúc và nội dung của một trang web.

Vai trò của HTML:

- Xác định cấu trúc trang web (tiêu đề, đoạn văn, bảng, hình ảnh,...).

- Làm việc chung với CSS (trang trí giao diện) và JavaScript (tương tác).

- Là cơ sở để các công nghệ như React, Angular,... hoạt động.

## 2.4 CSS

CSS (viết tắt của *Cascading Style Sheets*) là ngôn ngữ định kiểu, dùng để trang trí, định dạng và bố cục giao diện cho các tài liệu HTML.

Mục đích của CSS:

- Tách biệt phần nội dung (HTML) và giao diện (CSS).

- Giúp quản lý kiểu dáng nhất quán cho toàn bộ website.

- Tăng khả năng **tái sử dụng** mã và **dễ bảo trì**.

Ưu điểm:

- Giao diện chuyên nghiệp, linh hoạt.

- Tái sử dụng mã dễ dàng.

- Tách nội dung và kiểu dáng → dễ bảo trì.

- Tối ưu hóa trải nghiệm người dùng (responsive, hiệu ứng,...).

Nhược điểm nếu không dùng CSS

- Trang web chỉ hiển thị nội dung "thô".

- Khó kiểm soát giao diện khi có nhiều trang.

- Giao diện không đồng nhất, kém thu hút.

## 2.5 JavaScript

JavaScript là một ngôn ngữ lập trình thông dịch (interpreted), hướng đối tượng, động (dynamic), chủ yếu được sử dụng để xây dựng các tính năng tương tác trên trang web. JavaScript được thiết kế để chạy trên trình duyệt, cho phép nhà phát triển thao tác nội dung HTML, điều khiển kiểu dáng (CSS), xử lý sự kiện người dùng và giao tiếp với máy chủ mà không cần tải lại toàn bộ trang.

Vai trò của JavaScript trong web là xử lý logic, tương tác người dùng, hiệu ứng động. Một số tính năng nổi bật của JavaScript là:

- Xử lý sự kiện: click chuột, nhập bàn phím, rê chuột,...

- Tương tác với HTML/CSS: thay đổi nội dung, màu sắc, ẩn/hiện thành phần,...

- Kiểm tra dữ liệu form: xác thực trước khi gửi.

- Giao tiếp với server: thông qua AJAX hoặc Fetch API.

- Tạo hiệu ứng động: slide, menu đổ xuống, popup,...

- Lập trình hướng đối tượng và mô-đun.

Ưu điểm:

- Tăng tính tương tác và trải nghiệm người dùng.

- Hỗ trợ xây dựng ứng dụng web động (Single Page Application).

- Được hỗ trợ bởi mọi trình duyệt hiện đại.

- Có thể chạy cả trên server thông qua Node.js.

Nhược điểm:

- Có thể bị tắt bởi người dùng (hiếm).

- Không bảo mật tuyệt đối: mã JS có thể bị xem và chỉnh sửa.

- Dễ bị lạm dụng gây lỗi hoặc chậm trang nếu không tối ưu.

## 2.6 jQuery

jQuery là một thư viện JavaScript nhỏ gọn, nhanh và đa nền tảng, được thiết kế để đơn giản hóa việc thao tác với HTML, xử lý sự kiện, hiệu ứng động và Ajax trên trang web.

jQuery giúp viết JavaScript dễ hơn, ngắn gọn hơn và tương thích tốt hơn với nhiều trình duyệt.

Ưu điểm:

- Viết ít, làm được nhiều.

- Tương thích tốt với nhiều trình duyệt cũ.

- Học nhanh cho người mới bắt đầu với JavaScript.

- Cộng đồng lớn, nhiều plugin hỗ trợ sẵn.

Nhược điểm:

- Kích thước file thêm vào trang web (nếu dùng không hợp lý).

- Một số tính năng đã được JavaScript hiện đại (ES6+) hỗ trợ gốc nên jQuery không còn cần thiết.

- Không phù hợp cho các dự án lớn, hiện đại (SPA)

## 2.7 Bootstrap

Bootstrap là một framework mã nguồn mở dùng để thiết kế giao diện web hiện đại, responsive (tương thích nhiều thiết bị). Nó bao gồm tập hợp sẵn các HTML, CSS và JavaScript được xây dựng theo chuẩn thiết kế thống nhất.

Tính năng nổi bật:

- Responsive: Tự động thích ứng kích thước màn hình.

- Dễ dùng: Chỉ cần gán class, không cần viết CSS nhiều.

- Thống nhất: Giao diện đồng bộ, hiện đại, theo chuẩn thiết kế UI/UX.

- Tùy biến cao: Có thể chỉnh sửa theo phong cách riêng.

Ưu điểm:

- Tăng tốc độ phát triển giao diện.

- Hạn chế lỗi hiển thị giữa các trình duyệt.

- Tài liệu đầy đủ, cộng đồng lớn, hỗ trợ tốt.

- Hàng trăm thành phần sẵn sàng dùng.

Nhược điểm:

- Giao diện dễ “na ná” các web khác nếu không tùy chỉnh lại.

- Nếu dùng nhiều mà không tối ưu → tăng kích thước trang.

- Phụ thuộc vào class → dễ loạn nếu dùng không hợp lý

## 2.8 Node.js

Node.js là một môi trường chạy JavaScript phía máy chủ (server-side), cho phép lập trình viên sử dụng JavaScript để xây dựng các ứng dụng phía server, chẳng hạn như API, hệ thống xử lý dữ liệu thời gian thực, và ứng dụng web.

Node.js cho phép chạy JavaScript không chỉ trong trình duyệt, mà còn trên máy chủ, mở ra khả năng dùng cùng một ngôn ngữ cho cả frontend và backend.

Kiến trúc của Node.js:

- Dựa trên mô hình event-driven (hướng sự kiện) và non-blocking I/O (I/O không đồng bộ).

- Phù hợp với ứng dụng thời gian thực: chat, game online, streaming, API server....

Ưu điểm:

- Hiệu năng cao, xử lý đồng thời tốt.

- Dễ học với người đã biết JavaScript.

- Cộng đồng lớn, thư viện phong phú (npm).

- Dễ xây dựng ứng dụng thời gian thực (real-time app).

- Sử dụng JavaScript ở cả frontend và backend.

Nhược điểm:

- Không phù hợp cho các tác vụ nặng về CPU (tính toán phức tạp).

- Gỡ lỗi đôi khi phức tạp hơn các môi trường đồng bộ.

- Gọi callback nhiều dễ gây “callback hell” nếu không quản lý tốt

## 2.9 Express.js

Express.js là một framework web nhẹ và linh hoạt cho Node.js, dùng để xây dựng API và ứng dụng web một cách nhanh chóng, dễ hiểu.

Ưu điểm:

- Đơn giản, nhẹ

- Xử lý route linh hoạt

- Tốc độ xử lý tốt

- Hệ sinh thái lớn

- Dễ mở rộng

Nhược điểm:

- Không có cấu trúc mặc định

- Quản lý middleware phức tạp

- Chưa phù hợp cho lập trình viên mới

## 2.10 Socket.io

Socket.IO là một thư viện cho phép giao tiếp thời gian thực hai chiều giữa client và server thông qua WebSocket.

Ưu điểm:

- Giao tiếp real-time

- Tự động fallback

- Dễ tích hợp với Express

- Hỗ trợ “room” và “namespace”

- Tự động reconnect

Nhược điểm:

- Không phải WebSocket "thuần"

- Chiếm tài nguyên nếu không tối ưu

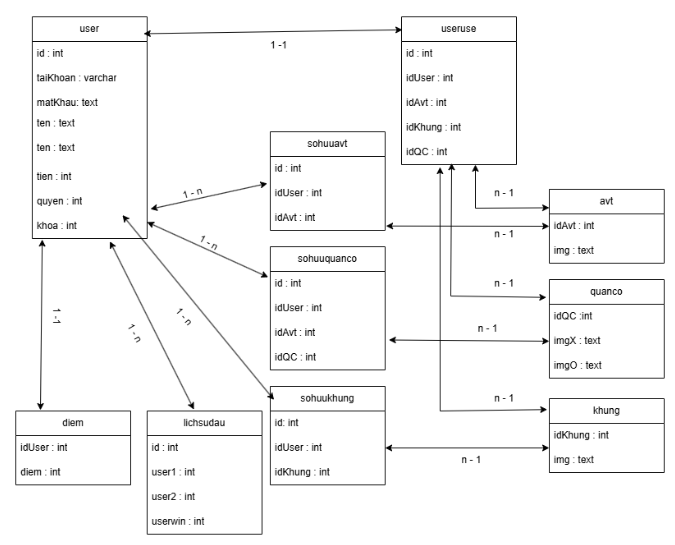
- Debug khó khi ứng dụng lớn

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 3.1 Mô hình quan niệm

Hình 3.1 Mô hình quan niệm

## 3.2 Mô hình luận lý



Hình 3.2 Mô hình luận lý

## 3.3 Mô hình vật lý

user ( **id**, taiKhoan, matKhau, ten, tien, quyen, khoa )

diem ( **idUser**, diem )

avt ( **idAvt**, img )

khung ( **idKhung**, img )

quanco ( **idQC** , imgX, imgO )

sohuuavt ( **id**, idUser, idAvt )

sohuukhung ( **id**, idUser, idKhung )

sohuuquanco ( **id**, idUser, idQC )

useruse ( **id**, idUser, idAvt, idKhung, idQC )

lichsudau (id, user1, user2, userwin)

## 3.4 Mô hình liên kết

Mô hình đạt dạng chuẩn 3NF vì toàn bộ các miền giá trị của các cột trong bảng đều chỉ chứa các giá trị nguyên tử (nguyên tố) và mỗi cột chỉ chứa một giá trị từ miền, các thuộc tính không khoá phải phụ thuộc hàm đầy đủ vào khoá chính và các thuộc tính không khoá phải phụ thuộc trực tiếp vào khoá chính.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.4 Mô hình liên kết

## 3.5 Mô hình xử lý (UML)

### A diagram of a computer AI-generated content may be incorrect.3.5.1 Mô hình xử lý nước đi

Hình 3.5.1 Mô hình xử lý nước đi

### A diagram with text and arrows AI-generated content may be incorrect.3.5.2 Mô hình xử lý chuyển đổi bàn cờ

Hình 3.5.2 Mô hình xử lý chuyển đổi bàn cờ

# CHƯƠNG 4: CHƯƠNG TRÌNH ỨNG DỤNG

## 4.1 Giao diện đăng nhập

Giao diện đăng nhập giúp người chơi truy cập vào website bằng cách nhập tên đăng nhập và mật khẩu. Khi đăng nhập thành công, người chơi có thể vào phòng chơi, thi đấu với người khác, và xem điểm số của mình.

A screenshot of a login box

AI-generated content may be incorrect.Nếu nhập sai thông tin, hệ thống sẽ báo lỗi và yêu cầu đăng nhập lại. Và nếu tài khoản của bạn đã bị khoá thì hệ thống sẽ thông báo tài khoản bạn đã bị khoá. Khi bạn không có tài khoản bạn có thể nhấn vào đường dẫn Đăng ký để đăng ký tài khoản.

Hình 3.1 Giao diện đăng nhập

## 4.2 Giao diện đăng ký

Giao diện đăng ký cho phép người chơi tạo tài khoản mới để sử dụng website. Người dùng cần cung cấp các thông tin như tên đăng nhập, mật khẩu, và tên người dùng. Sau khi đăng ký thành công, tài khoản sẽ được lưu vào cơ sở dữ liệu và người chơi có thể sử dụng để đăng nhập vào hệ thống.

A screenshot of a chat

AI-generated content may be incorrect.Hệ thống sẽ kiểm tra các thông tin đầu vào, như: tên đăng nhập và tên người dùng đã tồn tại hay chưa,... Nếu có lỗi, hệ thống sẽ hiển thị thông báo để người dùng sửa lại.

Hình 4.2 Giao diện đăng ký

## 4.3 Giao diện trang chủ

A screenshot of a chat

AI-generated content may be incorrect.Trang chủ là trung tâm điều hướng chính của website, nơi người dùng có thể tương tác, tạo phòng và tham gia các ván đấu.

Hình 4.3.1 Giao diện trang chủ

Trang chủ chia giao diện thành 2 phần:

* Phần bên trái gồm:

A screenshot of a computer

AI-generated content may be incorrect.+ Danh sách Online: hiển thị tất cả người chơi đang online và có thể mời họ vào chơi nếu bạn đang ở trong phòng.

Hình 4.3.2 Giao diện danh sách người chơi online

A screenshot of a phone

AI-generated content may be incorrect.+ Bảng xếp hạng: hiển thị xếp hạng có những người chơi top đầu và xếp hạng của chính người chơi đó.

Hình 4.3.3 Giao diện bảng xếp hạng

A screenshot of a chat

AI-generated content may be incorrect. + Chat tổng: cho phép người dùng có thể chat với tất cả người chơi đang online.

Hình 4.3.4 Giao diện chat tổng

* Phần bên phải gồm:

+ Thanh điêu hướng bên trên cho phép người dùng có thể quay về trang chủ và trở lại các phòng đang ở. Ô màu xanh là phòng đang ở, còn màu vàng nâu có nghĩa là đang trong trận và đối thủ vừa đánh cờ và tới lượt của bạn.

A yellow and green squares with black numbers

AI-generated content may be incorrect.

Hình 4.3.5 Giao diện thanh điều hướng

A screenshot of a chat

AI-generated content may be incorrect. + Chuông: khi click vào sẽ hiện những lời mời mà người chơi khác mời mình.

Hình 4.3.6 Giao diện chuông và lời mời chơi

A close-up of a screen

AI-generated content may be incorrect. + Nút tạo phòng: khi click vào sẽ hiện lên form tạo phòng để điền thông tin phòng như: tên phòng và phòng có mật khẩu không.

Hình 4.3.7 Giao diện tạo phòng

A white background with black lines

AI-generated content may be incorrect. + Danh sách phòng: hiển thị các phòng hiện đang có và từng phòng có những nút: “Vào lại” khi mình đang trong phòng đó, “Đã đầy” phòng đó đã đủ người chơi, “Vào” thì có thể vào phòng bình thường, “Nhập mật khẩu” phòng đó yêu cầu nhập mật khẩu để vào.

Hình 4.3.8 Danh sách phòng

## 4.4 Giao diện phòng

A screenshot of a computer

AI-generated content may be incorrect.Khi người chơi tạo hoặc tham gia vào phòng thì giao diện phòng sẽ xuất hiện và cho phép người dùng sử dụng các tính năng như: Thêm bot, Sẵn sàng, Rời phòng, Chat phòng,…

Hình 4.4 Giao diện phòng

## 4.5 Giao diện game

A screenshot of a game

AI-generated content may be incorrect. Khi toàn bộ người chơi đã sẵn sàng thì giao diện game sẽ hiện lên và trò chơi bắt đầu. Giao diện gồm bàn cờ 10x10, avatar 2 người chơi cùng với tên, chat phòng, 1 dòng thông báo lượt và 1 nút đầu hàng.

Hình 4.5 Giao diện game

## 4.6 Giao diện chế độ thường trực

Chế độ thường trực này cho phép màn hình hiển thị toàn bộ bàn cờ đang đánh tối đa là 4 bàn cờ để theo dõi và chơi cùng lúc tất cả bàn cờ.

A screenshot of a game

AI-generated content may be incorrect.

Hình 4.6 Giao diện chế độ thường trực

## 4.7 Giao diện chỉnh sửa thông tin cá nhân

### 4.7.1 Đổi avatar

A screenshot of a computer

AI-generated content may be incorrect.Người dùng có thể nhận và đổi avatar tại đây.

Hình 4.7.1 Giao diện đổi avatar

### 4.7.2 Đổi khung

A screenshot of a computer

AI-generated content may be incorrect.Người dùng có thể nhận và đổi khung tại đây.

Hình 4.7.2 Giao diện đổi khung

### 4.7.3 Đổi quân cờ

A screenshot of a computer

AI-generated content may be incorrect.Người dùng có thể nhận và đổi quân cờ tại đây.

Hình 4.7.3 Giao diện đổi quân cờ

### 4.7.4 Đổi thông tin cá nhân

Người dùng có thể đổi tên người dùng và mật khẩu ở đây.

A screenshot of a chat

AI-generated content may be incorrect.

Hình 4.7.4.1 Giao diện đổi tên người dùng

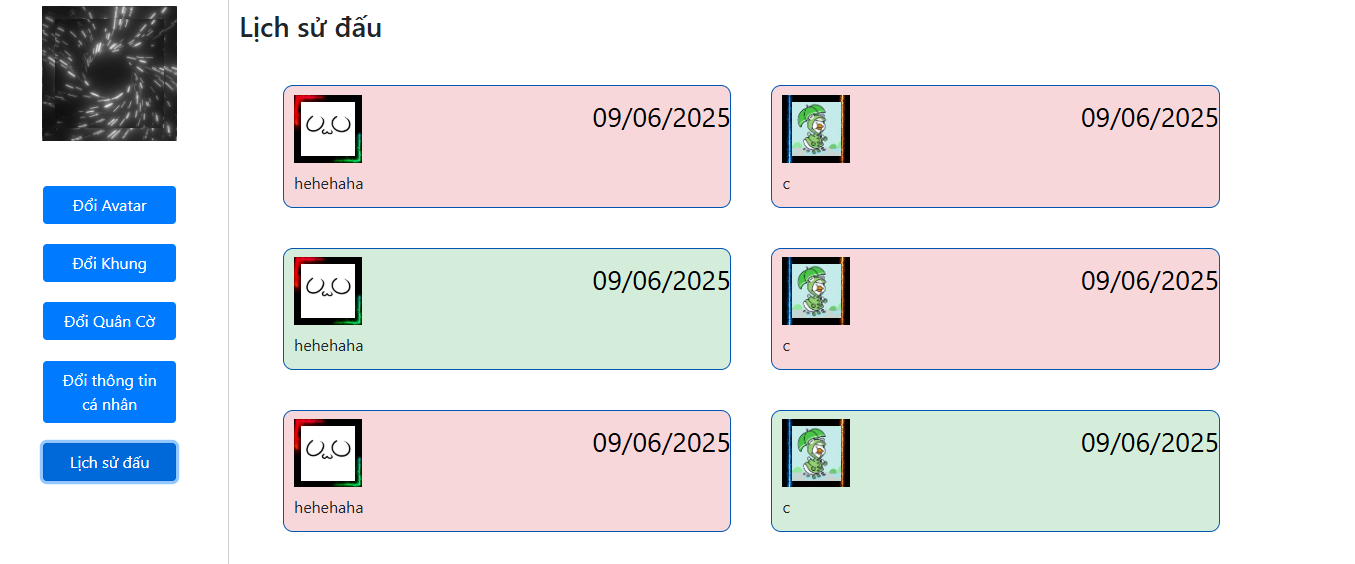
A screenshot of a chat

AI-generated content may be incorrect.

Hình 4.7.4.2 Giao diện đổi mật khẩu

### 4.7.5 Lịch sử đấu

Người dùng có thể xem lịch sử đấu với người của mình.



Hình 4.7.5 Giao diện lịch sử đấu

## 4.8 Giao diện Admin

### 4.8.1 Quản lý người dùng

Người quản trị có thể xem, thêm, sửa, xoá người dùng.

A screenshot of a computer

AI-generated content may be incorrect.  
Hình 4.8.1.1 Giao diện quản lý người dùng

A screenshot of a chat

AI-generated content may be incorrect.

Hình 4.8.1.2 Giao diện thêm,sửa người dùng

### 4.8.2 Quản lý avatar

A screenshot of a computer

AI-generated content may be incorrect.Người quản trị có thể xem, thêm, sửa, xoá avatar.

Hình 4.8.2.1 Giao diện quản lý avatar

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.8.2.2 Giao diện thêm, sửa avatar

### 4.8.3 Quản lý khung

A screenshot of a computer

AI-generated content may be incorrect.Người quản trị có thể xem, thêm, sửa, xoá khung.

Hình 4.8.3.1 Giao diện quản lý khung

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.8.3.2 Giao diện thêm, sửa khung

### 4.8.4 Quản lý quân cờ

A screenshot of a computer

AI-generated content may be incorrect.Người quản trị có thể xem, thêm, sửa, xoá quân cờ.

Hình 4.8.4.1 Giao diện quản lý quân cờ

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.8.4.2 Giao diện thêm, sửa quân cờ

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1 Kết luận

Sau quá trình nghiên cứu, thiết kế và triển khai, website chơi cờ caro online với ít nhất 2 người đã hoàn thiện với các chức năng cơ bản đáp ứng nhu cầu giải trí và thi đấu trực tuyến giữa các người chơi. Ứng dụng được xây dựng trên nền tảng web, sử dụng kiến trúc Client-Server kết hợp với công nghệ WebSocket để xử lý các tương tác thời gian thực, đảm bảo tính ổn định và mượt mà trong quá trình chơi.

Trong suốt quá trình phát triển, hệ thống đã tích hợp đầy đủ các chức năng cần thiết như: đăng ký, đăng nhập, tạo phòng chơi, mời người chơi, trò chuyện. Bên cạnh đó, các chức năng quản lý người dùng như thêm, sửa, xóa, phân quyền cũng được thiết kế để dễ dàng mở rộng và quản trị hệ thống trong tương lai.

Việc thực hiện đề tài này không chỉ giúp người phát triển rèn luyện kỹ năng lập trình fullstack (HTML, CSS, JavaScript, Node.js, MySQL...) mà còn hiểu rõ hơn về cách tổ chức một hệ thống web hoàn chỉnh theo mô hình MVC, đảm bảo sự tách biệt rõ ràng giữa phần giao diện, xử lý và dữ liệu. Ngoài ra, việc xử lý kết nối real-time bằng Socket.io đã mang lại trải nghiệm thực tế về cách hoạt động của các ứng dụng tương tác trực tuyến.

Mặc dù hệ thống đã hoàn thiện ở mức cơ bản, nhưng trong quá trình xây dựng vẫn còn nhiều giới hạn về giao diện, khả năng mở rộng và tối ưu hóa hiệu năng. Tuy nhiên, đây là nền tảng vững chắc để tiếp tục phát triển thêm nhiều tính năng mới trong tương lai.

## 5.2 Hướng phát triển

Để hoàn thiện hơn trong tương lai, hệ thống đánh cờ caro online có thể được nâng cấp và mở rộng với những định hướng sau:

* Lưu chi tiết và phân tích trận đấu.
* Tăng cường bảo mật và chống gian lận
* Tích hợp tính năng bạn bè và nhắn tin riêng
* Tích hợp chức năng thanh toán mua các vật phẩm

# TÀI LIỆU KHAM KHẢO

[1] Node.js. (n.d.), "HTTP: response.socket", https://nodejs.org/docs/latest/api/http.html#responsesocket

[2] Trung tâm Đào tạo Công nghệ Khoa Phạm. (2017, May 12), "Build web real time với NODEJS + SOCKETIO" [Video], YouTube. <https://www.youtube.com/watch?v=ofptoO93IFI&list=PLaDCaCbr3vmlmCC84sCzhHySIi73pipwV>

[3] Socket.IO. (n.d.), "Step 7 – Sharing messages", <https://socket.io/docs/v4/tutorial/step-7>

[4] Socket.IO. (n.d.), "Step 8 – Broadcasting events", <https://socket.io/docs/v4/tutorial/step-8>

[5] Socket.IO. (n.d.), "Step 5 – Emitting events", https://socket.io/docs/v4/tutorial/step-5