**BÁO CÁO SẢN PHẨM UETegame**

****

*Đỗ Minh Nhật - Nguyễn Quý Đang - Nguyễn Nhật Minh - Vũ Vân Long*

**MỤC LỤC**

| **PHẦN I. THÔNG TIN CHUNG** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Thông tin thành viên | | | | | 3 |
| 2 | Phân công công việc | | | | | 3 |
| **PHẦN II. GIỚI THIỆU CHUNG** | | | | | | |
| 1 | Tổng quan công nghệ sử dụng | | | | | 4 |
| 2 | Tổng quan hệ thống | | | | | 6 |
| **PHẦN III. CHI TIẾT HỆ THỐNG** | | | | | | |
| 1 | Thiết kế API | | | | | 12 |
| 2 | Đặc tả các tính năng chính của hệ thống | | | | | 16 |
| 3 | Module AI | | | | | 35 |
| **PHẦN IV: DEMO SẢN PHẨM** | | | | | | |
| 1 | Cách chạy sản phẩm | | | | | 39 |
| 2 | Phần chơi game | | | | | 39 |
| **PHẦN V: FUTURE WORK** | | | | | | |
| **PHẦN VI: NGUỒN, TÀI LIỆU THAM KHẢO** | | | | | | |

# PHẦN I - THÔNG TIN CHUNG

**Thông tin thành viên:**

Đỗ Minh Nhật

* Mã sinh viên: 22022537
* Lớp: AI2, Viện trí tuệ nhân tạo, trường Đại học Công nghệ, ĐHQGHN
* Email: 22022537@vnu.edu.vn

Nguyễn Quý Đang

* Mã sinh viên:22022500
* Lớp: AI2, Viện trí tuệ nhân tạo, trường Đại học Công nghệ, ĐHQGHN
* Email: 22022500@vnu.edu.vn

Vũ Vân Long

* Mã sinh viên: 22022501
* Lớp: AI2, Viện trí tuệ nhân tạo trường Đại học Công nghệ, ĐHQGHN
* Email: 22022501@vnu.edu.vn

Nguyễn Nhật Minh

* Mã sinh viên: 22022503
* Lớp: AI2, Viện trí tuệ nhân tạo trường Đại học Công nghệ, ĐHQGHN
* Email: 22022503@vnu.edu.vn

**Phân công công việc**

Đỗ Minh Nhật

* Thiết kế giao diện, thiết kế API backend
* Cài đặt giao diện bằng Next.js
* Cài đặt phần game bằng Phaser.js
* Cài đặt một số phần Backend
* Cài đặt Firebase
* Tham gia viết report

Nguyễn Quý Đang

* Cài đặt các API của backend
* Nối các API backend vào frontend
* Tham gia viết report
* Nghĩ ra những kịch bản xấu và nêu phương hướng áp dụng
* Viết Dockerfile

Vũ Vân Long

* Làm model AI, finetune mô hình, chuẩn bị code
* Viết README.md
* Kiểm tra code, nối model AI vào các nhánh chính
* Tham gia viết websocket
* Tham gia viết report

Nguyễn Nhật Minh

* Tham gia viết report chính
* Viết README.md
* Thiết kế luồng CI/CD, actions
* Cài đặt một số phần langchain
* Tham gia viết websocket

# PHẦN II - GIỚI THIỆU CHUNG

**GIỚI THIỆU**

Ứng dụng UETegame là một ứng dụng học tập tương tác, giúp người dùng học bài thông qua việc biến bài học thành các trò chơi thú vị. Người dùng có thể tạo phòng chơi bằng cách tải lên tài liệu học và từ đó tạo ra các trò chơi giáo dục, giúp tăng cường sự hấp dẫn và hiệu quả trong việc học tập.

**CÔNG NGHỆ SỬ DỤNG**

****

| Next.js không chỉ là một framework React thông thường mà còn là một nền tảng hoàn chỉnh cho việc xây dựng ứng dụng web hiệu quả. Với khả năng tối ưu hóa SEO, Server-Side Rendering (SSR), Client-Side Rendering (CSR), và cơ chế tối ưu hóa tải trang, Next.js giúp tăng cường trải nghiệm người dùng và hiệu suất ứng dụng. Ngoài ra, việc sử dụng TypeScript cùng với Next.js cung cấp tính linh hoạt và kiểm soát mã nguồn mạnh mẽ. |
| --- |



| React là một thư viện JavaScript mạnh mẽ được phát triển bởi Facebook, tập trung vào việc xây dựng giao diện người dùng linh hoạt và dễ bảo trì. Với việc sử dụng các components tái sử dụng và Virtual DOM, React giúp tối ưu hóa hiệu suất ứng dụng và quản lý trạng thái ứng dụng một cách hiệu quả. Hơn nữa, cộng đồng lớn và sự hỗ trợ mạnh mẽ từ các công cụ như Redux làm cho React trở thành một trong những lựa chọn hàng đầu cho việc phát triển ứng dụng web hiện đại. |
| --- |



| LangChain là một khung phần mềm mã nguồn mở được thiết kế để đơn giản hóa việc phát triển các ứng dụng trí tuệ nhân tạo (AI) sử dụng các mô hình ngôn ngữ lớn (LLM). Nó cung cấp một bộ sưu tập các thành phần và công cụ giúp các nhà phát triển dễ dàng xây dựng các ứng dụng như chatbot, tóm tắt văn bản, trả lời câu hỏi và tạo nội dung sáng tạo. |
| --- |



| Python là một ngôn ngữ lập trình phổ biến với cú pháp đơn giản và dễ hiểu, là lựa chọn hàng đầu cho nhiều loại ứng dụng từ phát triển web đến phân tích dữ liệu và Machine Learning. Với một cộng đồng lớn và sự hỗ trợ mạnh mẽ từ các thư viện như NumPy, Pandas và TensorFlow, Python là công cụ mạnh mẽ cho việc phát triển các dự án công nghệ. |
| --- |

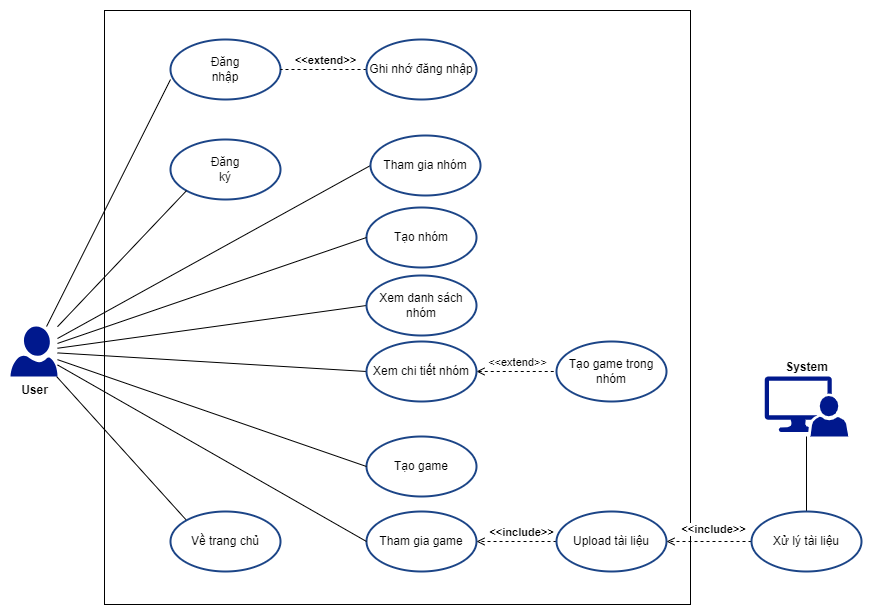


| Docker là một nền tảng container mã nguồn mở cho việc xây dựng, vận chuyển và triển khai ứng dụng một cách dễ dàng và linh hoạt. Bằng cách đóng gói ứng dụng và tất cả các phụ thuộc của nó vào các container độc lập, Docker giúp đơn giản hóa quá trình phát triển và triển khai ứng dụng. Với tính di động và khả năng tương thích với môi trường khác nhau, Docker là công cụ lý tưởng cho việc xây dựng và chạy các ứng dụng phức tạp trong môi trường sản xuất |
| --- |



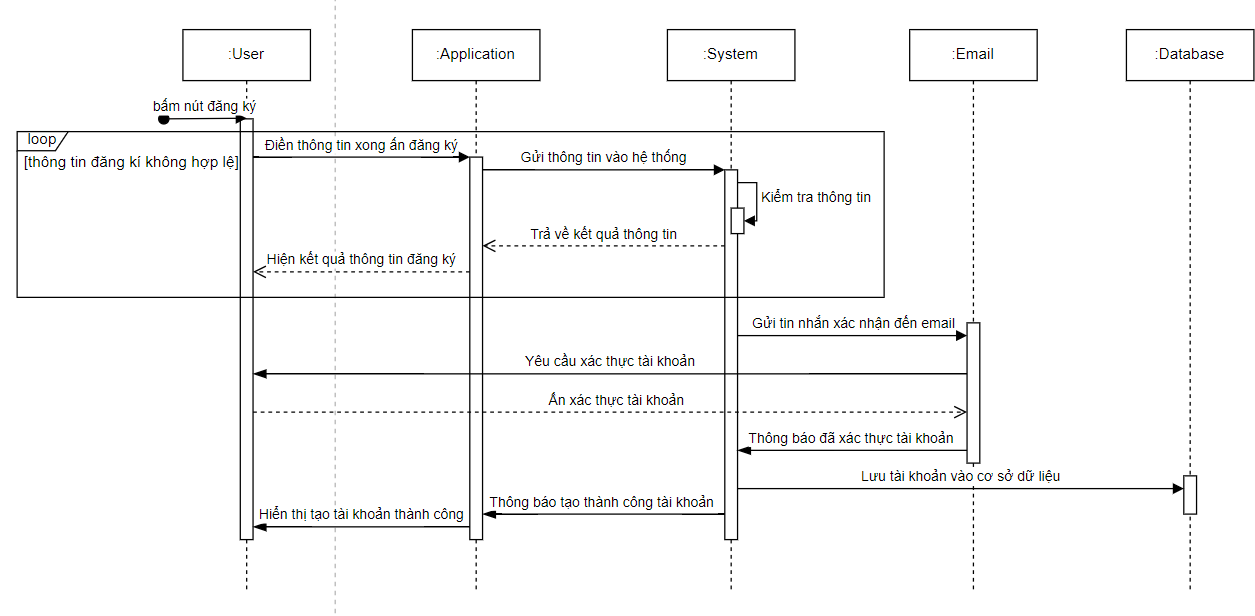
| Firebase Database là một dịch vụ cơ sở dữ liệu NoSQL thời gian thực được cung cấp bởi Google. Firebase Database là dịch vụ cơ sở dữ liệu NoSQL thời gian thực do Google cung cấp. Nó nổi bật giữa các giải pháp lưu trữ dữ liệu khác nhờ những ưu điểm vượt trội sau:  **1. Realtime Updates - Cập nhật dữ liệu tức thì:**  Điểm đặc biệt của Firebase Database là khả năng đồng bộ dữ liệu **thời gian thực**. Bất kỳ thay đổi nào được thực hiện trên dữ liệu sẽ được phản ánh ngay lập tức trên tất cả các thiết bị được kết nối, mang đến trải nghiệm ứng dụng mượt mà và năng động. Tính năng này cực kỳ hữu ích cho các ứng dụng đòi hỏi sự tương tác liên tục giữa người dùng, ví dụ như trò chuyện, bảng tin tức, hay theo dõi vị trí.  **2. Offline Access - Truy cập dữ liệu mọi lúc mọi nơi:**  Khả năng **truy cập ngoại tuyến** là một ưu điểm không thể bỏ qua của Firebase Database. Ngay cả khi thiết bị mất kết nối internet, người dùng vẫn có thể truy cập và thao tác với dữ liệu được lưu trữ cục bộ. Khi thiết bị kết nối trở lại, dữ liệu sẽ được tự động đồng bộ hóa với database. Nhờ vậy, ứng dụng của bạn luôn sẵn sàng phục vụ người dùng, bất kể họ đang ở đâu hay tình trạng kết nối mạng như thế nào.  **3. NoSQL Structure - Dữ liệu linh hoạt, dễ dàng mở rộng:**  Firebase Database sử dụng cấu trúc dữ liệu **NoSQL** dựa trên JSON. JSON là định dạng dữ liệu linh hoạt, dễ dàng lưu trữ và xử lý các dạng dữ liệu phức tạp. Nhờ vậy, bạn có thể dễ dàng mở rộng quy mô ứng dụng mà không cần lo lắng về việc thay đổi cấu trúc database.  **4. Security - Bảo mật toàn diện:**  Firebase Database được tích hợp sẵn hệ thống xác thực Firebase Authentication, giúp bảo vệ dữ liệu khỏi truy cập trái phép. Bạn có thể kiểm soát quyền truy cập dữ liệu dựa trên danh tính người dùng hoặc các quy tắc truy cập tùy chỉnh. Đảm bảo an toàn cho thông tin nhạy cảm là ưu tiên hàng đầu của Firebase, giúp bạn xây dựng ứng dụng uy tín và tin cậy.  **5. Scalability - Khả năng mở rộng cao:**  Firebase Database được thiết kế để **mở rộng** theo nhu cầu của bạn. Khi lượng người dùng và dữ liệu tăng lên, hệ thống sẽ tự động điều chỉnh để đáp ứng nhu cầu xử lý, đảm bảo hiệu suất hoạt động ổn định và mượt mà. Bạn không cần lo lắng về việc ứng dụng bị gián đoạn hay quá tải dữ liệu.  **6. Developer-friendly - Dễ dàng sử dụng và tích hợp:**  Firebase Database cung cấp các API đơn giản, dễ sử dụng và tích hợp với nhiều nền tảng di động phổ biến như Android, iOS, React Native, v.v. Nhờ vậy, các nhà phát triển có thể nhanh chóng xây dựng và triển khai ứng dụng mà không cần tốn nhiều thời gian và công sức.  **7. Cost-effective - Hiệu quả về chi phí:**  Firebase Database cung cấp **gói miễn phí** với mức dung lượng dữ liệu và lưu lượng truy cập hợp lý, phù hợp cho các dự án khởi nghiệp hoặc ứng dụng có lượng người dùng vừa phải. Các gói trả phí với mức dung lượng cao hơn cũng sẵn có để đáp ứng nhu cầu của các ứng dụng lớn hơn. |
| --- |

**TỔNG QUAN HỆ THỐNG**

****

*Hình 2.1: Sơ đồ Use Case*

Ở đây khi vào ứng dụng, điều cơ bản nhất user có thể làm chính là Đăng nhập hoặc Đăng ký. Mỗi user cũng có thể tạo nhóm, tham gia nhóm, xem danh sách nhóm mà mình nằm trong (Gọi là group) để tăng tính tương tác; ngoài ra từng User cũng có thể tạo và tham gia game. Về cơ bản 1 game được tạo nên khi người dùng upload file câu hỏi lên trên server, module AI của hệ thống sẽ xử lý tài liệu và tạo thành một loạt các câu hỏi. Kế đến hệ thống sẽ tạo một màn chơi game đánh quái từ các câu hỏi trên (Để tiêu diệt quái bạn phải trả lời đúng, nếu trả lời sai sẽ bị mất máu)



*Hình 2.2 Sơ đồ tuần tự của hệ thống đăng ký*

**Mô tả**

**Bước 1:** Người dùng điền thông tin đăng ký vào giao diện và nhấn nút “Đăng ký”.

**Bước 2:** Từ giao diện, thông tin đăng ký được gửi vào hệ thống.

**Bước 3:** Hệ thống kiểm tra thông tin đăng ký.

**Bước 4:** Hệ thống trả về kết quả kiểm tra thông tin cho ứng dụng.

**Bước 5:** Ứng dụng hiển thị kết quả kiểm tra thông tin đăng ký cho người dùng. Nếu thông tin không hợp lệ, người dùng sẽ phải nhập lại thông tin.

**Bước 6:** Nếu thông tin hợp lệ, hệ thống gửi yêu cầu xác thực tài khoản bằng cách gửi một tin nhắn xác nhận đến email của người dùng.

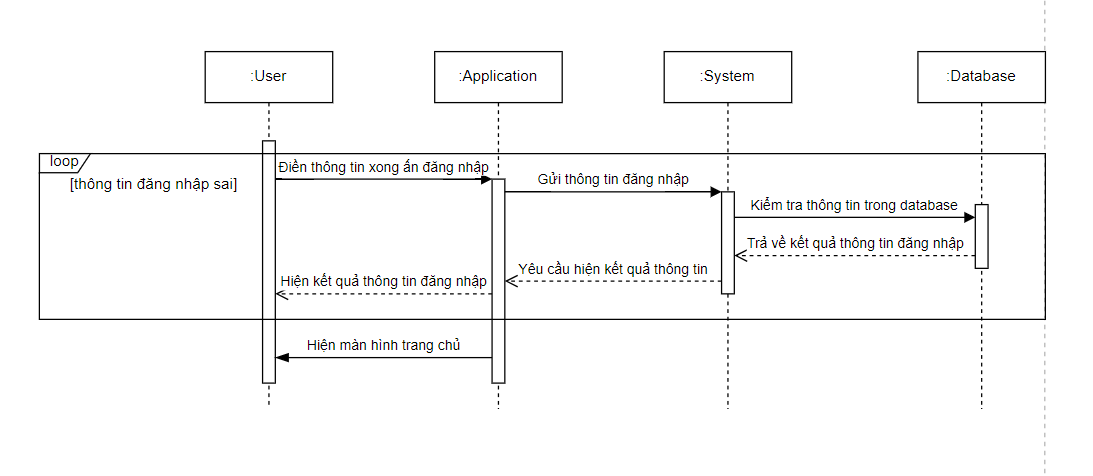
**Bước 7:** Hệ thống gửi tin nhắn xác nhận đến email của người dùng.

**Bước 8:** Người dùng nhấn vào liên kết trong email để xác thực tài khoản.

**Bước 9:** Hệ thống nhận được xác nhận từ người dùng và thông báo rằng tài khoản đã được xác thực thành công.

**Bước 10:** Hệ thống lưu thông tin tài khoản đã xác thực vào cơ sở dữ liệu.

**Bước 11:** Ứng dụng hiển thị thông báo rằng tài khoản đã được tạo thành công.



*Hình 2.3 Sơ đồ tuần tự của hệ thống đăng nhập*

**Mô tả**

**Bước 1:** Người dùng điền thông tin đăng nhập vào giao diện và nhấn nút “Đăng nhập”.

**Bước 2:** Từ giao diện, thông tin đăng nhập được gửi vào hệ thống.

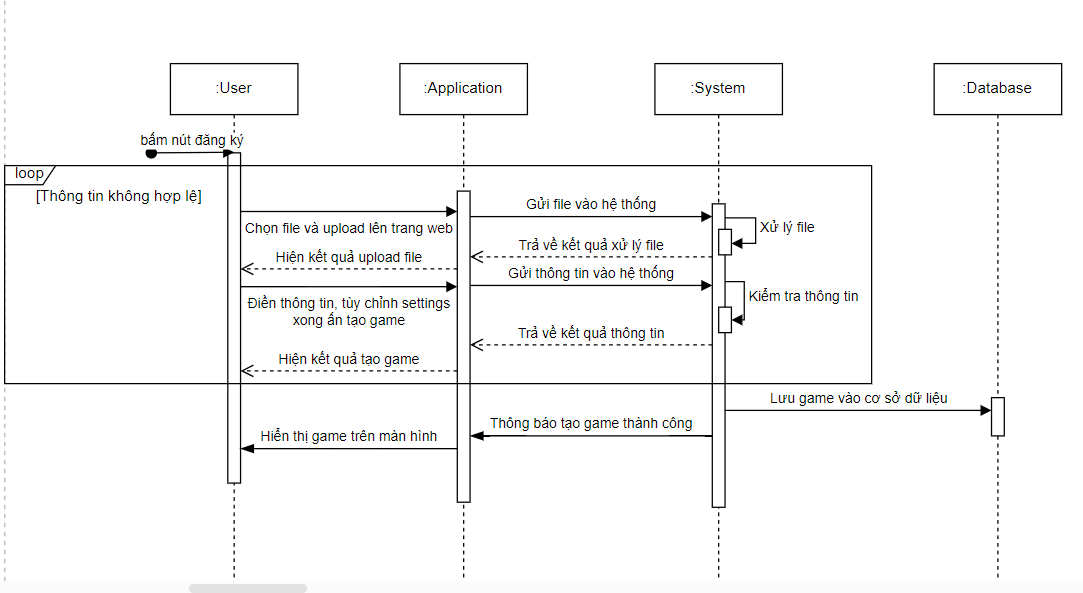
**Bước 3:** Hệ thống kiểm tra thông tin đăng nhập trong cơ sở dữ liệu.

**Bước 4:** Cơ sở dữ liệu trả về kết quả kiểm tra thông tin đăng nhập cho hệ thống.

**Bước 5:** Hệ thống yêu cầu hiển thị kết quả thông tin đăng nhập.

**Bước 6:** Ứng dụng hiển thị kết quả thông tin đăng nhập cho người dùng. Nếu thông tin đăng nhập sai, người dùng sẽ phải nhập lại thông tin.

**Bước 7:** Nếu thông tin đăng nhập đúng, ứng dụng hiển thị màn hình trang chủ cho người dùng.



*Hình 2.4 Sơ đồ tuần tự của hệ thống tạo game*

**Mô tả**

**Bước 1:** Người dùng chọn file và upload lên trang web.

**Bước 2:** Ứng dụng gửi file vào hệthống.

**Bước 3:** Hệ thống xử lý file.

**Bước 4:** Hệ thống trả về kết quả xử lý file cho ứng dụng.

**Bước 5:** Ứng dụng hiển thị kết quả upload file cho người dùng.

**Bước 6:** Người dùng điền thông tin và tùy chỉnh settings, sau đó nhấn nút tạo game.

**Bước 7:** Ứng dụng gửi thông tin vào hệ thống.

**Bước 8:** Hệ thống kiểm tra thông tin.

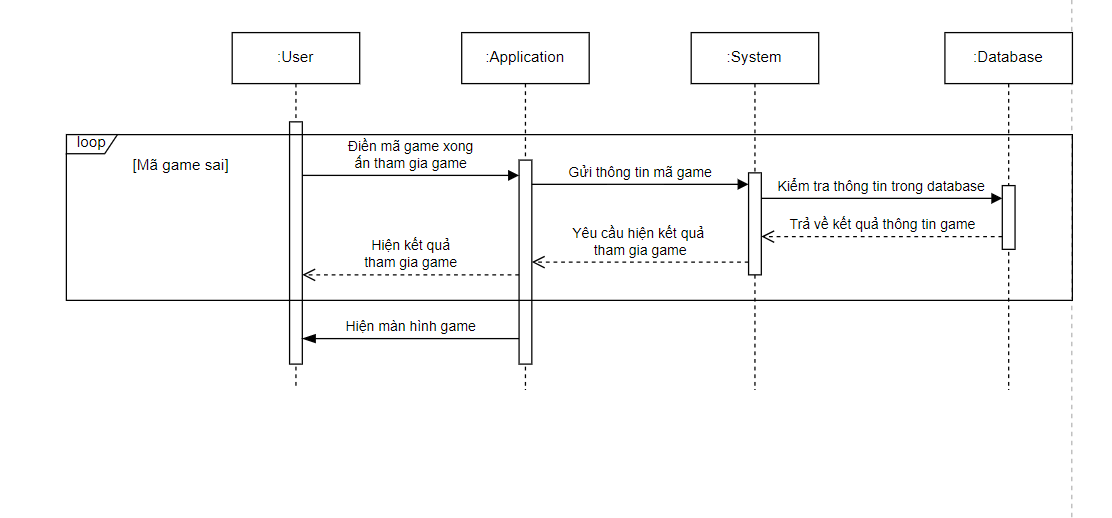
**Bước 9:** Hệ thống trả về kết quả thông tin cho ứng dụng.

**Bước 10:** Ứng dụng hiển thị kết quả tạo game cho người dùng.

**Bước 11:** Hệ thống lưu game vào cơ sở dữ liệu**.**

**Bước 12:** Ứng dụng thông báo tạo game thành công cho người dùng.

**Bước 13:** Ứng dụng hiển thị game trên màn hình.



*Hình 2.4 Sơ đồ tuần tự của hệ thống vào game*

**Mô tả**

**Bước 1:** Người dùng điền mã game và ấn tham gia game.

**Bước 2:** Ứng dụng gửi thông tin mã game vào hệ thống.

**Bước 3:** Hệ thống yêu cầu kiểm tra thông tin mã game từ cơ sở dữ liệu.

**Bước 4:** Cơ sở dữ liệu kiểm tra thông tin.

**Bước 5:** Cơ sở dữ liệu trả về kết quả thông tin game cho hệ thống.

**Bước 6:** Hệ thống trả về kết quả thông tin game cho ứng dụng.

**Bước 7:** Ứng dụng hiển thị kết quả tham gia game cho người dùng.

**Bước 8:** Ứng dụng hiển thị màn hình phòng game.

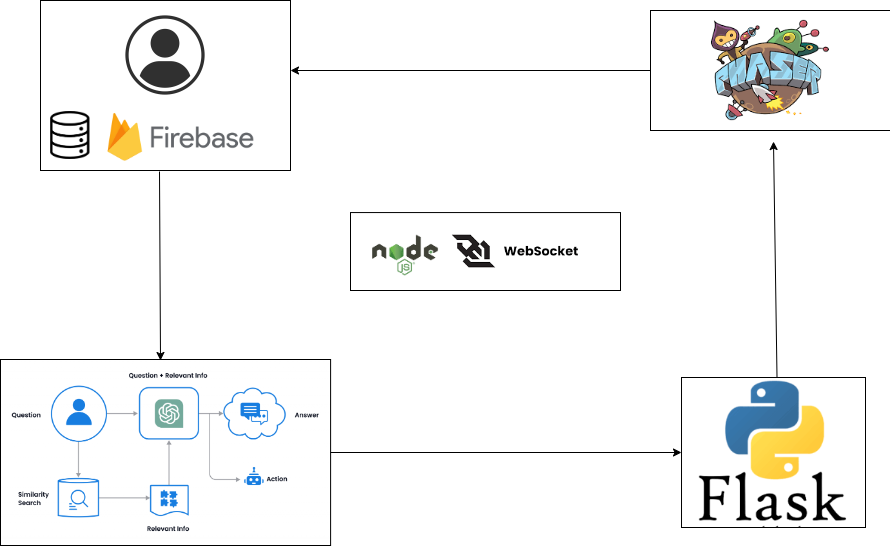


*Hình 2.5 Sơ đồ tuần tự quay về trang chính*

**Mô tả**

**Bước 1:** Người dùng ấn vào logo hoặc "Trang chủ".

**Bước 2:** Ứng dụng hiển thị trang chủ cho người dùng.



*Hình 2.2: Tổng quan hệ thống*

**1. Client:**

* **Chức năng:**
  + Hiển thị giao diện người dùng (UI) cho người chơi, bao gồm các yếu tố như màn hình trò chơi, thanh công cụ, bảng điểm, v.v.
  + Thu thập thông tin đầu vào từ người chơi, chẳng hạn như thao tác chuột, bàn phím hoặc màn hình cảm ứng.
  + Giao tiếp với Module AI và Module Nodejs thông qua API để truy cập dữ liệu trò chơi và tính năng AI.
* **Công nghệ:**
  + HTML, CSS, JavaScript: để phát triển giao diện người dùng.
  + Thư viện JavaScript dành cho game engine như Phaser hoặc Pixi.js: để xử lý đồ họa và tương tác người chơi.
  + Firebase JavaScript SDK: để quản lý đăng nhập, lưu trữ dữ liệu người dùng và đồng bộ hóa dữ liệu giữa các thiết bị.

**2. Module AI:**

* **Chức năng:**
  + Phân tích dữ liệu người chơi, bao gồm hành vi chơi game, sở thích và tiến trình.
  + Dựa trên phân tích, đề xuất các hành động tiếp theo cho người chơi, chẳng hạn như gợi ý vật phẩm, nâng cấp hoặc thử thách phù hợp.
  + Tạo ra các nhân vật phản diện và thử thách được cá nhân hóa cho từng người chơi, mang lại trải nghiệm chơi game độc đáo và hấp dẫn.
* **Công nghệ:**
  + Flask: Framework Python nhẹ và linh hoạt để xây dựng API.
  + Langchain: Thư viện Python để phát triển các mô hình học máy và xử lý ngôn ngữ tự nhiên.

**3. Module Nodejs:**

* **Chức năng:**
  + Là bộ phận cốt lõi của Backend, chịu trách nhiệm logic trò chơi và quản lý trạng thái trò chơi.
  + Xử lý các hành động của người chơi, bao gồm di chuyển nhân vật, sử dụng vật phẩm và tấn công kẻ thù.
  + Cập nhật trạng thái trò chơi theo thời gian thực và đồng bộ hóa dữ liệu với tất cả người chơi.
  + Giao tiếp với Module AI để lấy các đề xuất và dữ liệu liên quan.
  + Cung cấp API cho Module Client truy cập các tính năng trò chơi.
* **Công nghệ:**
  + Node.js: Môi trường runtime JavaScript cho phát triển Backend.
  + Socket.IO: Thư viện JavaScript để giao tiếp Websocket realtime.
  + Cơ sở dữ liệu NoSQL như MongoDB hoặc Redis: để lưu trữ dữ liệu trò chơi.

**4. Game code:**

* **Chức năng:**
  + Hiển thị đồ họa trò chơi 2D, bao gồm nhân vật, cảnh quan và các hiệu ứng.
  + Xử lý các sự kiện người chơi như di chuyển chuột, nhấp chuột và nhấn phím.
  + Giao tiếp với Module Nodejs để cập nhật trạng thái trò chơi và nhận dữ liệu trò chơi.
* **Công nghệ:**
  + Phaser: Thư viện JavaScript phổ biến để phát triển các trò chơi 2D và HTML5.
  + WebGL: Giao diện lập trình ứng dụng (API) JavaScript để truy cập khả năng đồ họa 3D của trình duyệt web.

**Lợi ích của hệ thống:**

* **Tính linh hoạt:** Kiến trúc mô-đun cho phép dễ dàng thêm các tính năng mới, mở rộng hệ thống và tích hợp với các dịch vụ bên ngoài.
* **Khả năng mở rộng:** Hệ thống có thể xử lý nhiều người chơi cùng lúc mà không gặp vấn đề về hiệu suất.
* **Tính an toàn:** Firebase cung cấp tính năng xác thực và lưu trữ dữ liệu an toàn, bảo mật thông tin người dùng.
* **Trí tuệ nhân tạo:** Module AI giúp mang lại trải nghiệm chơi game cá nhân hóa và hấp dẫn hơn.
* **Realtime:** Websocket đảm bảo đồng bộ hóa dữ liệu tức thời giữa tất cả người chơi, tạo ra trải nghiệm chơi game mượt mà.

# PHẦN III - CHI TIẾT HỆ THỐNG

**Thiết kế API Backend**

Về cơ bản sản phẩm gồm 6 Module chính đó là:

1. User Module: Chịu trách nhiệm xử lý các tác vụ ảnh hưởng trực tiếp tới user như lấy thông tin từ một user, thêm xóa sửa thông tin của user, …
2. Room Module: Phụ trách tất cả logic liên quan đến việc tạo, xóa, vào game room
3. Game Module (Sử dụng WebSocket): Toàn bộ logic phần game nằm gọn trong đây
4. Shop Module: Chịu trách nhiệm cho việc xử lý các giao dịch của người dùng
5. Group Module: Chịu trác nhiệm cho tất cả các quá trình xử lý liên quan tới các group game
6. Storage Module: Chịu trách nhiệm liên quan đến lưu trữ kho hàng và số dư tài khoản của user

**NGUYÊN TẮC THIẾT KẾ**

Toàn bộ sản phẩm được thiết kế theo các nguyên tắc

**Rule 1**: Method của Request phải thể hiện hành động của nó sẽ làm và dữ liệu trong body thường sử dụng **JSON** là format (đôi chỗ dùng form-data)

**Rule 2**: Response - Status code luôn được trả về với đúng status của response

**Rule 3**: **Idempotent** - Mỗi request thành công hay thất bại không phụ thuộc vào số lần được gọi

**Rule 4**: **Stateless** - Các request giữa một client với server độc lập và không ảnh hưởng đến nhau

**Rule 5**: Pagination - các request dạng **GET** khi trả về lượng lớn item thì không trả về toàn bộ item mà sử dụng offset&limit

***Hạn chế***

Phần code của sản phẩm này vi phạm một rule khá quan trọng đó chính là naming khi tên của API mặc dù được group theo từng chủ thể của hệ thống nhưng lại dùng động từ chỉ hành động để đặt tên API (Ví dụ: POST /room/get hay POST /room/update). Ngoài ra các API không có đánh dấu version ở đầu nên nếu scale ứng dụng về sau sẽ gặp nhiều khó khăn.

VD: /room/v3/users

Tuy nhiên theo nhóm phát triển thì điều này là hoàn toàn chấp nhận được bởi thời gian làm phần mềm tương đối ngắn không có thời gian đào sâu chi tiết vào việc áp dụng các tiêu chuẩn một cách triệt để nhất

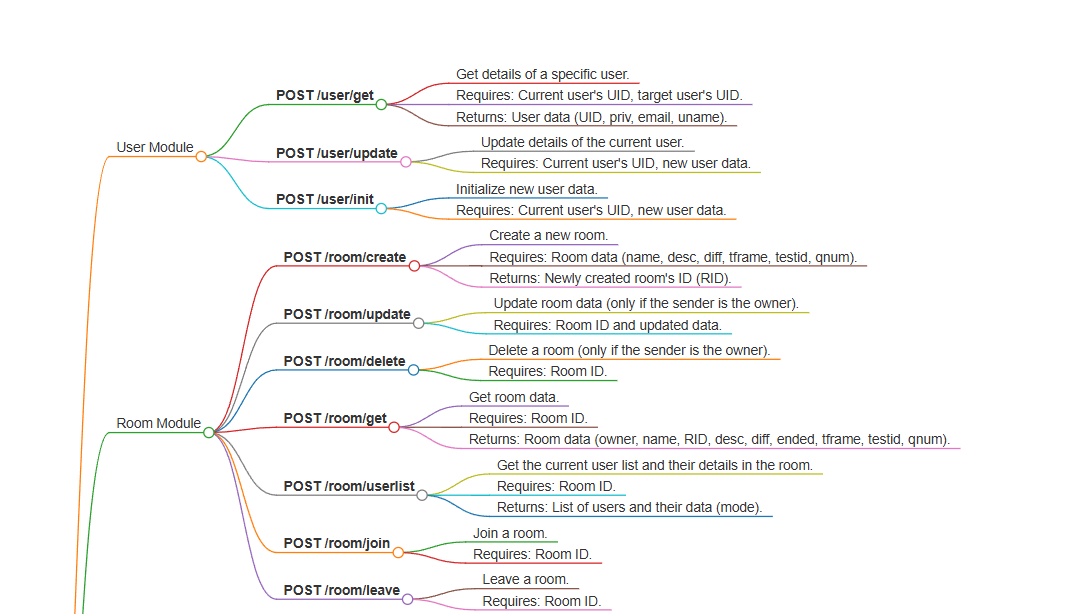
**ĐI SÂU VÀO CÁC MODULE BACKEND CHÍNH**

Ở đây để thuận tiện cho việc minh họa, nhóm đã sử dụng sơ đồ tư duy (được render bằng Markmap) cho có cái nhìn tổng quan về tất cả các luồng API trong hệ thống (Hình 3.1)



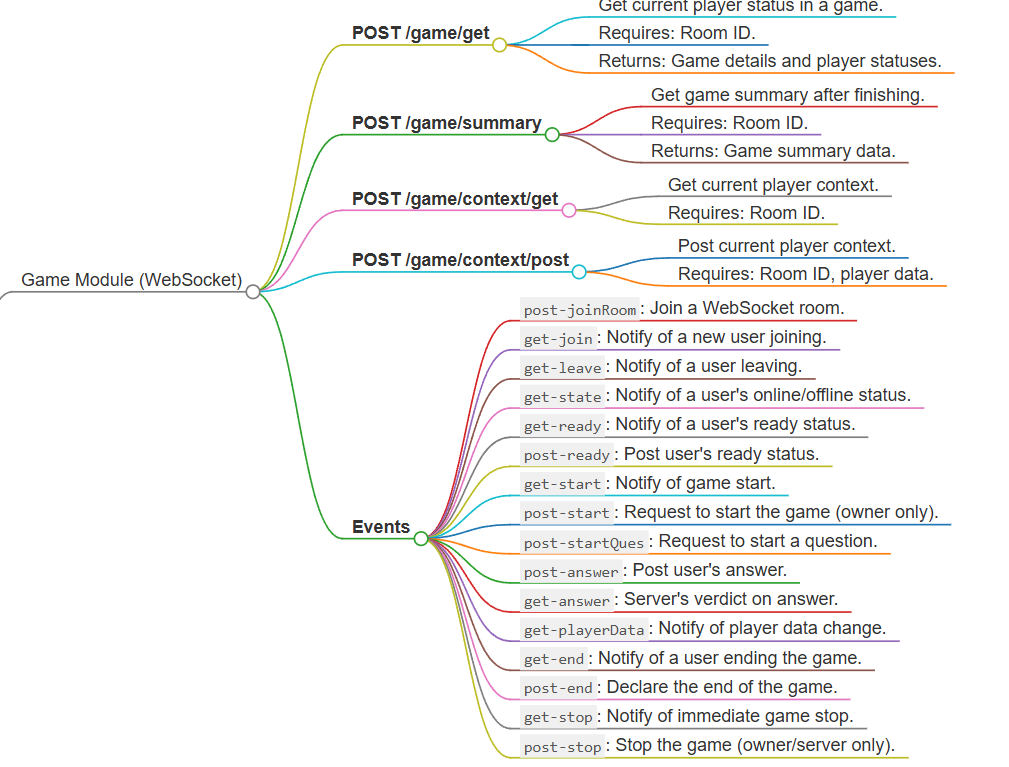
*Hình 3.1: API design cả hệ thống*

Giờ ta sẽ đi sâu vào từng module của hình trên, khởi đầu với User Module và RoomModule (Hình 3.2)



*Hình 3.2: User + Room Module API*

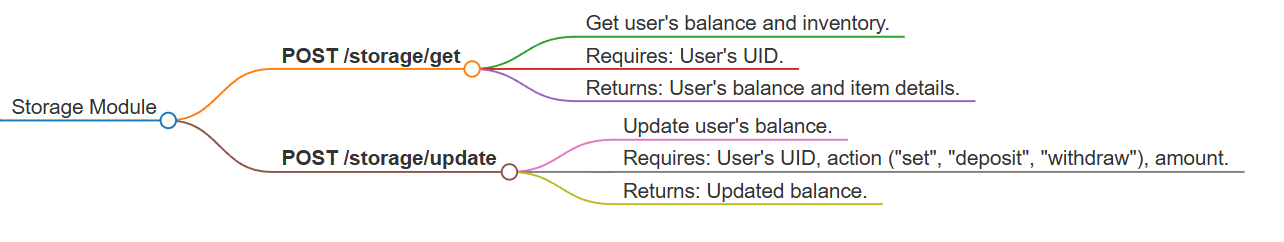
Tiếp đó ta xét đến các luồng API của Game Module (Websocket), ở đây ta lưu ý luồng Events sử dụng Websocket thay vì HTTP Request như bình thường để xử lý trò chơi trong thời gian thực (real-time) - hình ảnh minh họa nằm ở hình 3.3



*Hình 3.3: Game Module API*

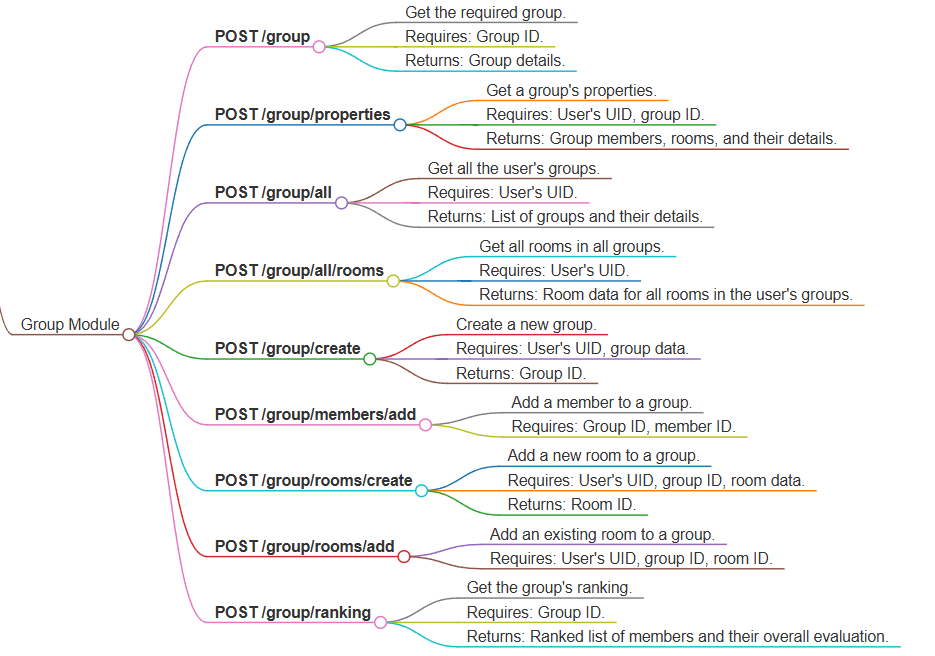
Tiếp đến ta xét thêm 2 luồng tương đối ngắn là Shop Module và Storage Module, cụ thể như sau





*Hình 3.4: Shop + Storage Module API*

Cuối cùng là Group Module với các chức năng chính liên quan mật thiết đến việc quản lý group game cụ thể



*Hình 3.4: Group Module API*

Có thể thấy Group Module gồm tương đối nhiều API do nó là một trong những trọng tâm mà nhóm phát triển hướng tới khi xây dựng ứng dụng (nhấn mạnh hai khía cạnh: “Hợp tác, cạnh tranh cùng phát triển”) vì thế Group Module là một phần không thể thiếu

| **Đăng nhập**  **Name:** Đăng nhập  **Trigger:** Người dùng bấm nút đăng nhập  **Precondition:** Người dùng đã có tài khoản và đang ở màn hình trang chủ của ứng dụng |
| --- |
| **Basic Path:**   1. Màn hình hiển thị trang đăng nhập, bao gồm các trường Tên tài khoản và Mật khẩu. 2. Người dùng nhập tên tài khoản và mật khẩu. 3. Người dùng bấm nút "Đăng nhập". 4. Hệ thống kiểm tra tính hợp lệ của thông tin đăng nhập. 5. Nếu thông tin hợp lệ, hệ thống cho phép người dùng truy cập vào ứng dụng.     **Alternative Paths:**  Nếu người dùng nhập sai tên tài khoản hoặc mật khẩu:   1. Hệ thống thông báo lỗi "Tên tài khoản hoặc mật khẩu không chính xác". 2. Người dùng được yêu cầu nhập lại thông tin đăng nhập.   **Postcondition:** Người dùng được đăng nhập vào ứng dụng và có thể truy cập các tính năng tương ứng với vai trò của mình. |

| **Đăng ký**  **Name:** Đăng ký  **Trigger:** Người dùng bấm nút đăng ký  **Precondition:** Người dùng chưa có tài khoản và đang ở màn hình trang chủ của ứng dụng |
| --- |
| **Basic Path:**   1. Người dùng bấm đăng ký và truy cập vào màn hình đăng ký. 2. Người dùng nhập thông tin cá nhân cần thiết để đăng ký tài khoản, email, mật khẩu. 3. Người dùng xác nhận thông tin đã nhập và bấm nút "Đăng ký". 4. Hệ thống kiểm tra tính hợp lệ của thông tin đăng ký. 5. Nếu thông tin hợp lệ, hệ thống tạo tài khoản mới cho người dùng và gửi email xác nhận đến địa chỉ email đã cung cấp.     **Alternative Paths:**  Nếu thông tin không hợp lệ:   1. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin cần thiết.   **Postcondition:**  Tài khoản mới được tạo thành công trên hệ thống. Người dùng có thể đăng nhập bằng thông tin đăng nhập mới tạo. |

| **Về trang chủ**  **Name:** Về trang chủ  **Trigger:** Người dùng bấm nút về trang chủ  **Precondition:** Người dùng đã đăng nhập vào ứng dụng |
| --- |
| **Basic Path:**   1. Người dùng bấm vào biểu tượng tài khoản. 2. Người dùng bấm vào nút "Về trang chủ". 3. Hệ thống chuyển hướng người dùng về màn hình trang chủ.     **Alternative Paths:**  Nếu người dùng nhập thiếu thông tin bắt buộc hoặc tên nhóm đã tồn tại:  Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin.  **Postcondition:**  Người dùng được chuyển về màn hình trang chủ của ứng dụng và có thể tiếp tục sử dụng các chức năng khác. |

| **Tạo nhóm**  **Name:** Tạo nhóm  **Trigger:** Người dùng bấm nút tạo nhóm  **Precondition:** Người dùng đã đăng nhập vào ứng dụng |
| --- |
| **Basic Path:**   1. Người dùng bấm vào nút "Tạo nhóm". 2. Màn hình hiển thị trang tạo nhóm, bao gồm các trường Tên nhóm và Mô tả nhóm. 3. Người dùng nhập tên nhóm và mô tả. 4. Người dùng bấm nút "Tạo nhóm". 5. Hệ thống kiểm tra tính hợp lệ của thông tin và tạo nhóm mới. 6. Hệ thống thông báo nhóm được tạo thành công.     **Alternative Paths:**  Nếu người dùng nhập thiếu thông tin bắt buộc hoặc tên nhóm đã tồn tại:   1. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin.   **Postcondition:**  Nhóm mới được tạo và lưu vào cơ sở dữ liệu, người dùng có thể thêm thành viên và bắt đầu sử dụng nhóm. |

| **Xem danh sách nhóm**  **Name:** Xem danh sách nhóm  **Trigger:** Người dùng bấm nút xem danh sách nhóm  **Precondition:** Người dùng đã đăng nhập vào ứng dụng |
| --- |
| **Basic Path:**   1. Người dùng bấm vào nút "Xem danh sách nhóm". 2. Hệ thống hiển thị danh sách các nhóm mà người dùng đã tham gia hoặc tạo.     **Alternative Paths:**  Không có  **Postcondition:**  Danh sách các nhóm liên quan được hiển thị cho người dùng. |

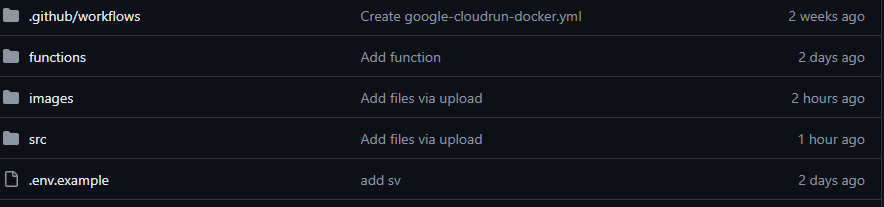
| **Tạo game**  **Name:** Tạo game  **Trigger:** Người dùng bấm nút tạo game  **Precondition:** Người dùng đã đăng nhập vào ứng dụng và đã bấm vào ảnh đại diện |
| --- |
| **Basic Path:**   1. Người dùng bấm vào nút "Tạo game". 2. Màn hình hiển thị trang tạo game, bao gồm các trường Tên game, Mô tả game. 3. Người dùng nhập tên game và mô tả. 4. Người dùng chọn template cho game, độ khó, thời gian cho từng câu hỏi. 5. Người dùng chọn loại tài liệu và tài liệu để upload và convert.      1. Hệ thống nhận file và convert file thành tài liệu cho game. 2. Người dùng bấm nút "Tạo game". 3. Hệ thống kiểm tra tính hợp lệ của thông tin và tạo game mới. 4. Hệ thống thông báo game được tạo thành công.     **Alternative Paths:**  Nếu người dùng nhập thiếu thông tin bắt buộc hoặc tên game đã tồn tại:   1. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin   **Postcondition:**  Game mới được tạo và lưu vào cơ sở dữ liệu, người dùng có thể bắt đầu chơi game hoặc mời người khác tham gia. |

| **Tham gia game**  **Name:** Tham gia game  **Trigger:** Người dùng bấm nút tham gia game  **Precondition:** Người dùng đã đăng nhập vào ứng dụng |
| --- |
| **Basic Path:**   1. Người dùng bấm vào nút "Tham gia game". 2. Hệ thống hiển thị màn hình có trường nhập code của phòng chơi. 3. Người dùng nhập code của trò chơi và bấm join. 4. Hệ thống trả về kết quả tham gia phòng chơi.     **Alternative Paths:**  Nếu không tồn tại phòng chơi có mã mà người dùng nhập vào:   1. Hệ thống trả về thông báo không tồn tại phòng chơi và yêu cầu người dùng nhập lại.   **Postcondition:**  Nếu tồn tại phòng chơi có mã mà người dùng nhập vào, người chơi sẽ tham gia vào phòng chơi đó. |

# PHẦN IV - DEMO SẢN PHẨM

**CÁCH CHẠY SẢN PHẨM**

Trước hết, khi vào phần source code ta sẽ thấy

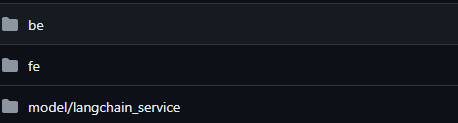


*Hình 4.1: Hình ảnh source code sản phẩm*

Toàn bộ phần source code chính sẽ ở mục src, trong mục này ta chú ý đến 3 thư mục chính:

1. **fe**: toàn bộ code front-end nằm trong phần này
2. **be**: toàn bộ code back-end nằm trong phần này
3. **model/langchain\_service**: toàn bộ code phần AI nằm trong phần này

Chi tiết nằm ở hình 4.2



*Hình 4.2: 3 thành phần chính của sản phẩm*

Tất nhiên ta phải clone code về trước bằng lệnh git clone; tiếp sau đó ta mở thư mục **fe** và chạy lệnh

**`npm install`** và **`npm run dev`** để phần front-end bắt đầu chạy ở <http://localhost:3000>

Kế tiếp ta cần vào thư mục **be**, lúc này thì có điểm cần lưu là ở đây ta phải đảm bảo file .env gồm có những trường sau:

* BACKEND\_PORT
* BACKEND\_WS\_GLOBAL\_EMIT
* BACKEND\_FLASK\_PORT
* BACKEND\_FLASK\_URI
* FRONTEND ORIGIN
* CACHE\_TTL AND CACHE\_CHECK
* ROOM\_ID\_LENGTH

sau khi tiến thành thủ tục trên ta tiếp tục chạy **`npm install`**  và **`npm run dev`**

Cuối cùng đến với phần model/langchain\_service, ta chỉ cần làm như sau:

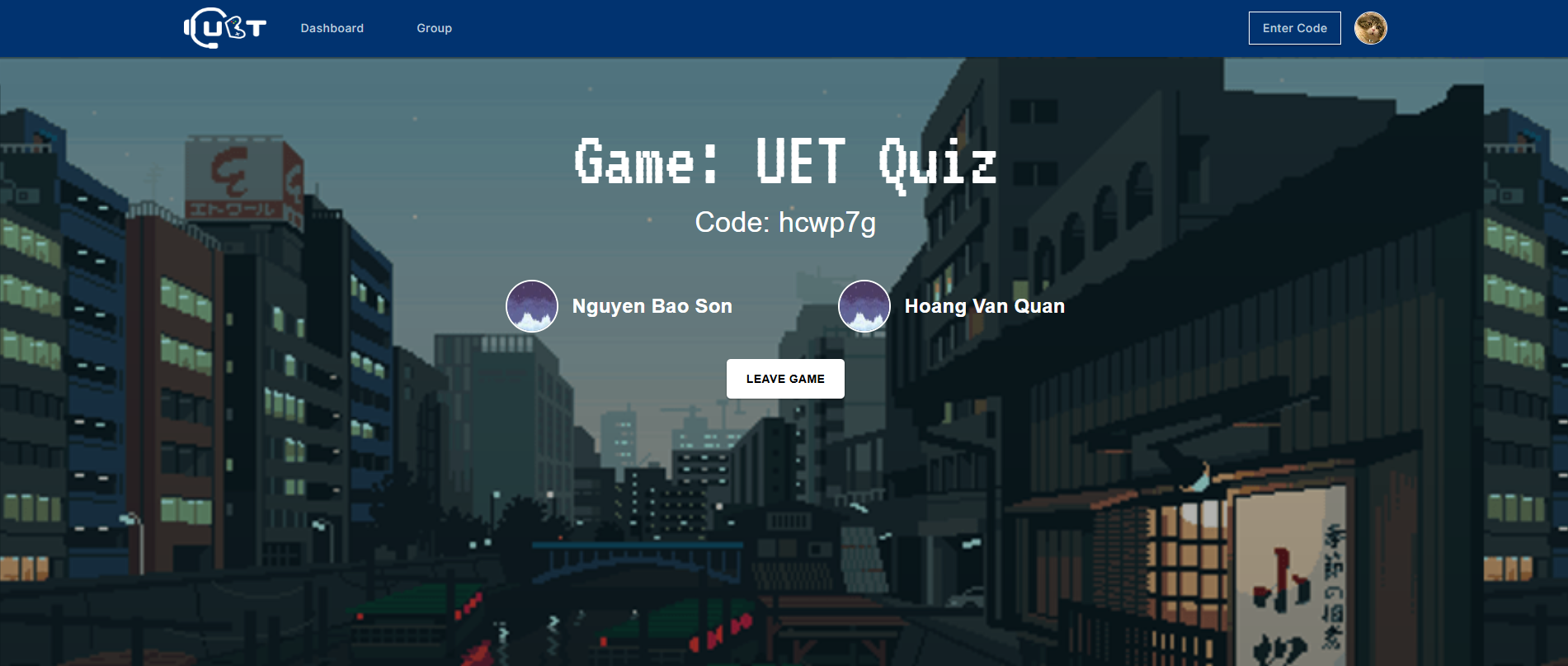
**`pip install -r requirements.txt`** rồi sau đó **`python server.py`**để chạy được các luồng Backend. Ngoài ra, ta cũng cần đảm bảo file .env gồm có:

* OPENAI\_API\_KEY
* OPENAI\_API\_BASE
* GOOGLE\_API\_KEY

**PHẦN TRÒ CHƠI**

**Phòng chơi:**

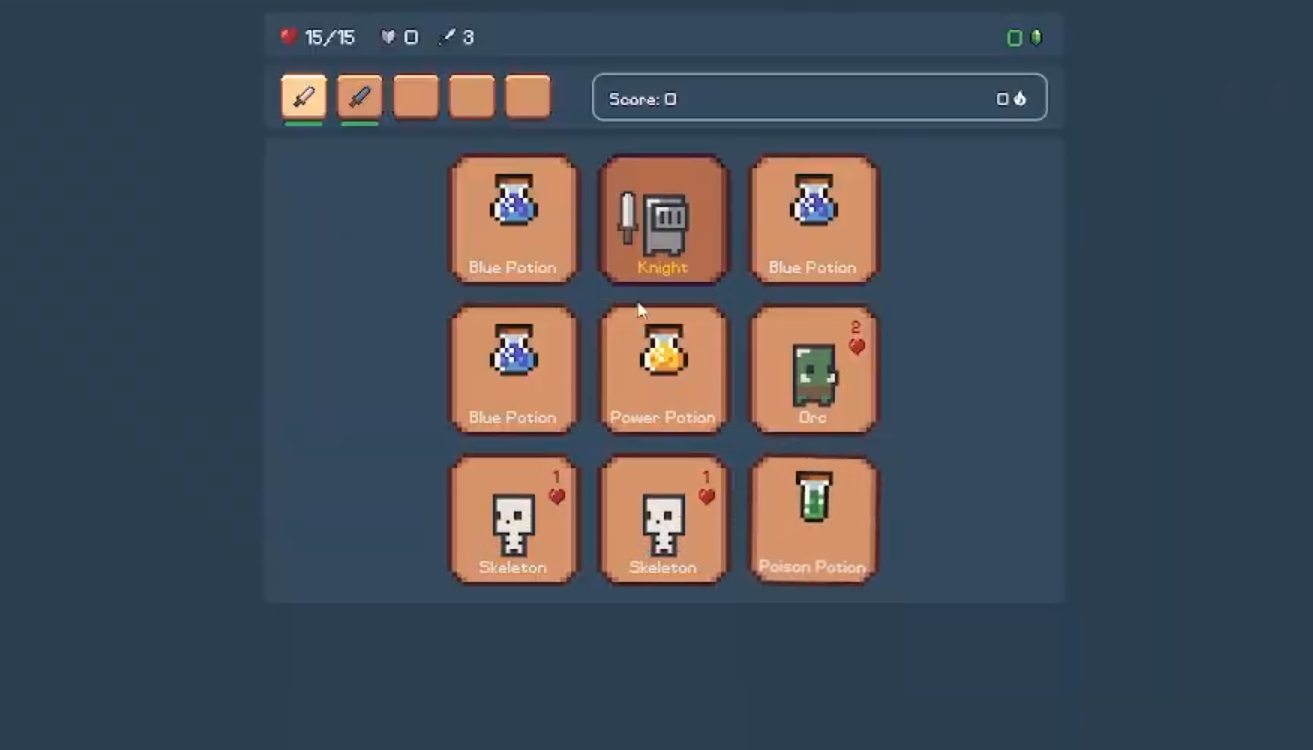
Người dùng có thể tạo phòng chơi và mời người dùng khác vào trong phòng chơi của mình.



**Di chuyển:**

Người chơi có thể chọn một trong hai cách sau để di chuyển nhân vật của mình trong game:

* Sử dụng các phím mũi tên Lên (↑), Xuống (↓), Trái (←), Phải (→) để di chuyển nhân vật.
* Sử dụng chuột và nhấn vào một trong 4 ô liền kề (Trên, Dưới, Phải, Trái) để di chuyển đến ô đó.



**Hiển thị:**

Ở phần trên của màn hình có một số biểu tượng:

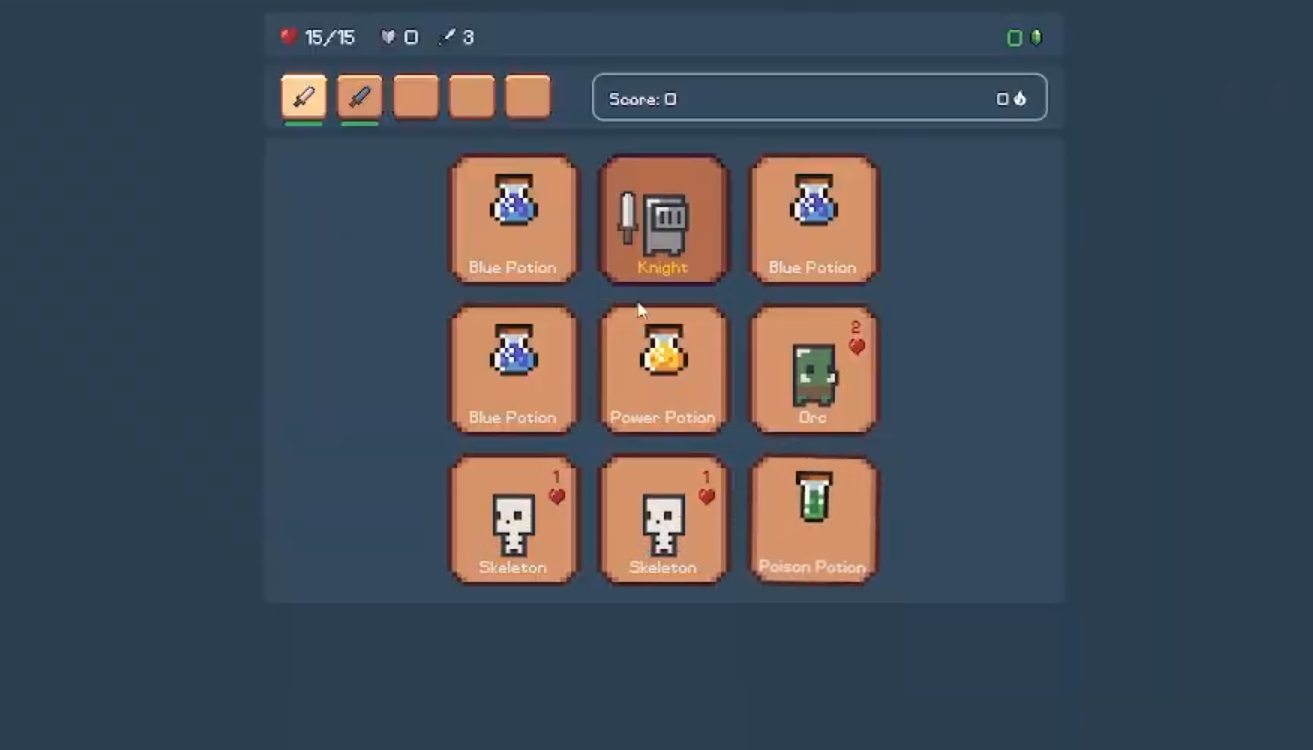
* Thanh máu: hiển thị số máu hiện tại của nhân vật, nếu máu của nhân vật giảm xuống 0, trò chơi sẽ kết thúc.
* Giáp: hiển thị chỉ số giáp của nhân vật, chỉ số giáp sẽ giúp giảm sát thương mà nhân vật của bạn sẽ nhận vào khi bị kẻ địch tấn công.

*(Mẹo: bạn có thể tăng chỉ số giáp của mình bằng cách trang bị khiên)*

* Kiếm: hiển thị chỉ số công kích của nhân vật, chỉ số công kích sẽ quyết định sát thương mà nhân vật của bạn gây ra khi tân công kẻ địch.

*(Mẹo: bạn có thể tăng chỉ số tấn công của mình bằng cách trang bị kiếm)*

* Điểm: số điểm hiện tại mà bạn đạt được (hãy cố gắng đạt điểm càng cao càng tốt)
* Kim cương: số kim cương bạn có trong túi, bạn có thể sử dụng kim cương để mua trang phục và vật phẩm cũng như nâng cấp chỉ số cơ bản của nhân vật.



**Hành động:**

Nhân vật của bạn sẽ có một số hành động tùy thuộc vào ô mà bạn di chuyển đến:

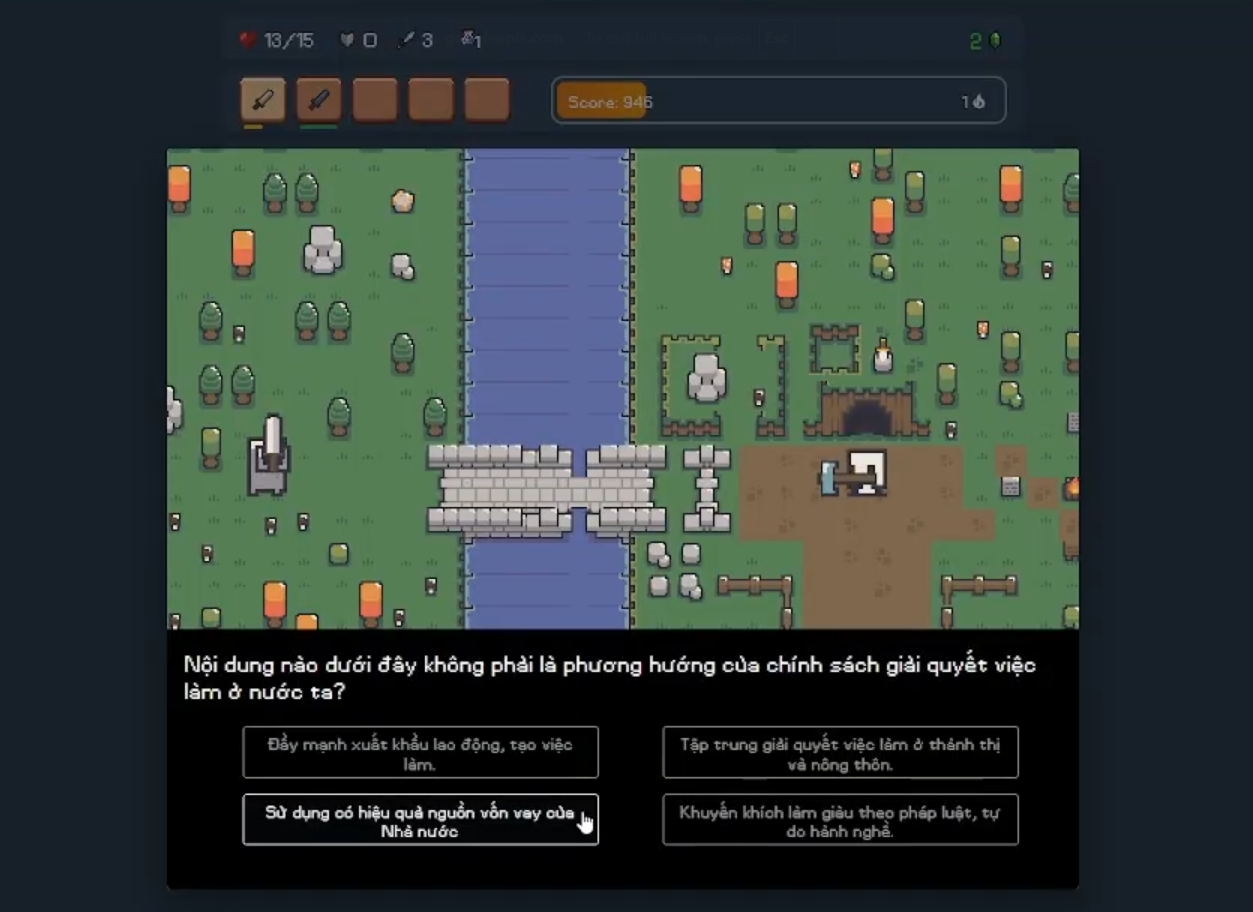
* Kim cương: nhân vật của bạn sẽ nhận được một kim cương. (Chúc mừng!!!)
* Kiếm: nhân vật của bạn sẽ nhận được một thanh kiếm trong túi.

*(Lưu ý: Kiếm của bạn sẽ bị hỏng sau một số lần sử dụng)*

* Khiên: nhân vật của bạn sẽ nhận được một cái khiên (chỉ số phòng ngự của bạn sẽ được tăng!!!)
* Bình thuốc:
* Thuốc xanh: Hãy cẩn thận! Khi đi vào ô này, nhân vật của bạn sẽ bị trừ máu
* Thuốc vàng: Sau khi uống thuốc này, nhân vật của bạn sẽ được cường hóa chỉ số tấn công.
* Thuốc đỏ: Hồi máu cho nhân vật của bạn

*(Lưu ý: bạn không thể hồi quá số máu tối đa. Hãy cân nhắc khi nào nên sử dụng vật phẩm hữu ích này)*

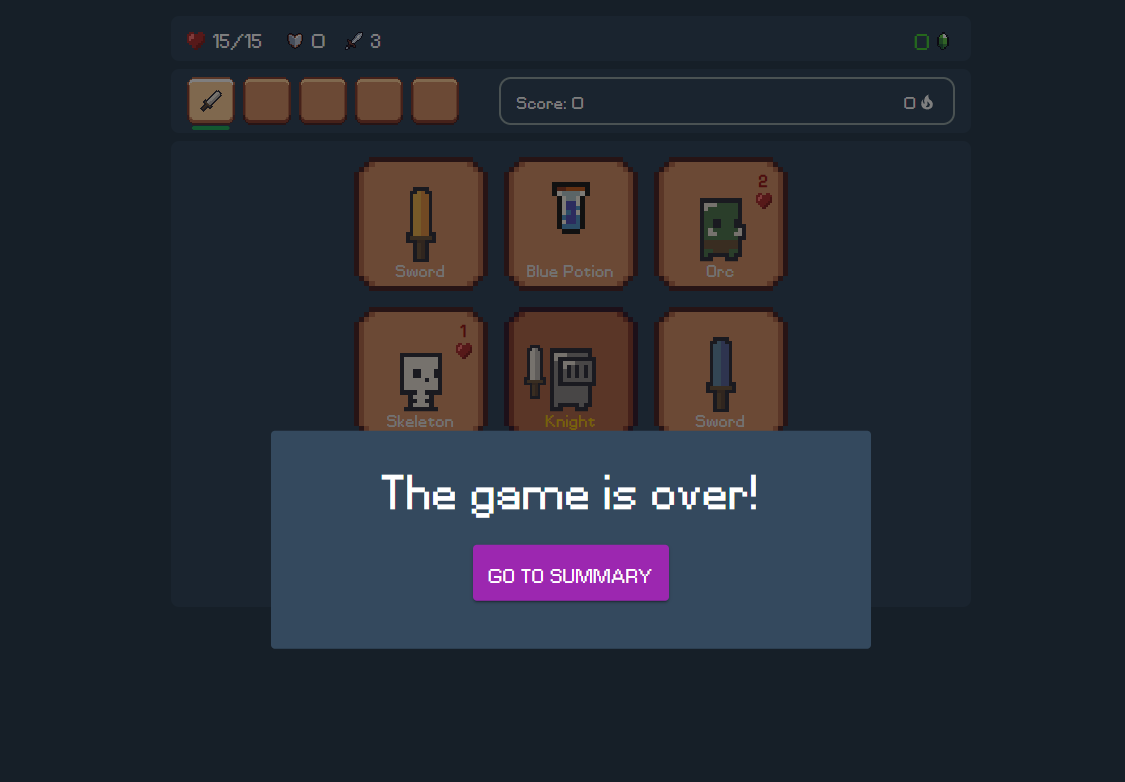
* Quái vật: Khi đi vào ô này, bạn sẽ phải chiến đấu với kẻ địch.
* Skeleton: Loại quái vật yếu nhất bạn có thể gặp phải.
* Orc: Loại quái vật này khỏe hơn những con Skeleton mà bạn từng chiến đấu.

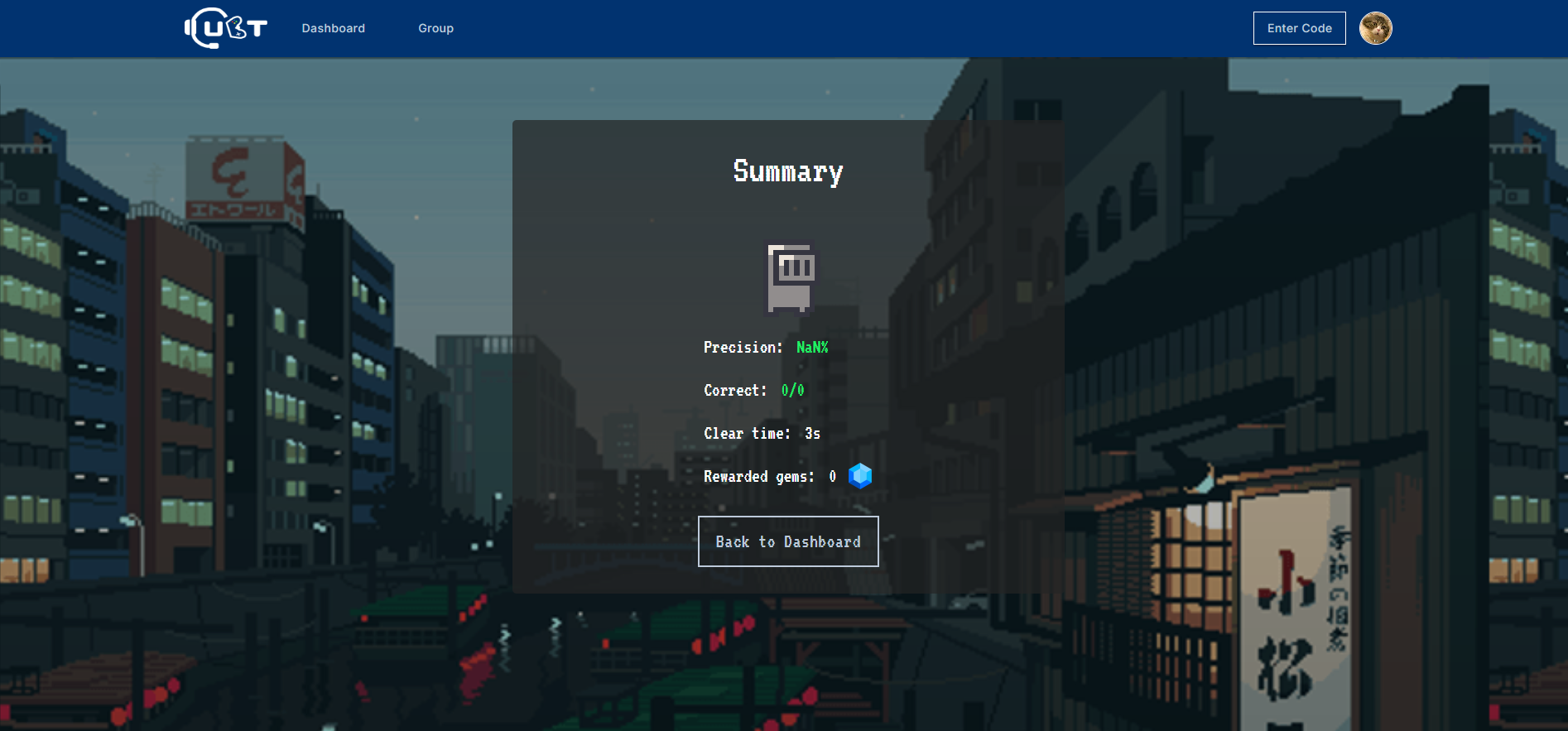


**Chiến đấu với quái vật**

Khi di chuyển vào ô quái vật, bạn sẽ chuyển đến màn hình đánh quái.

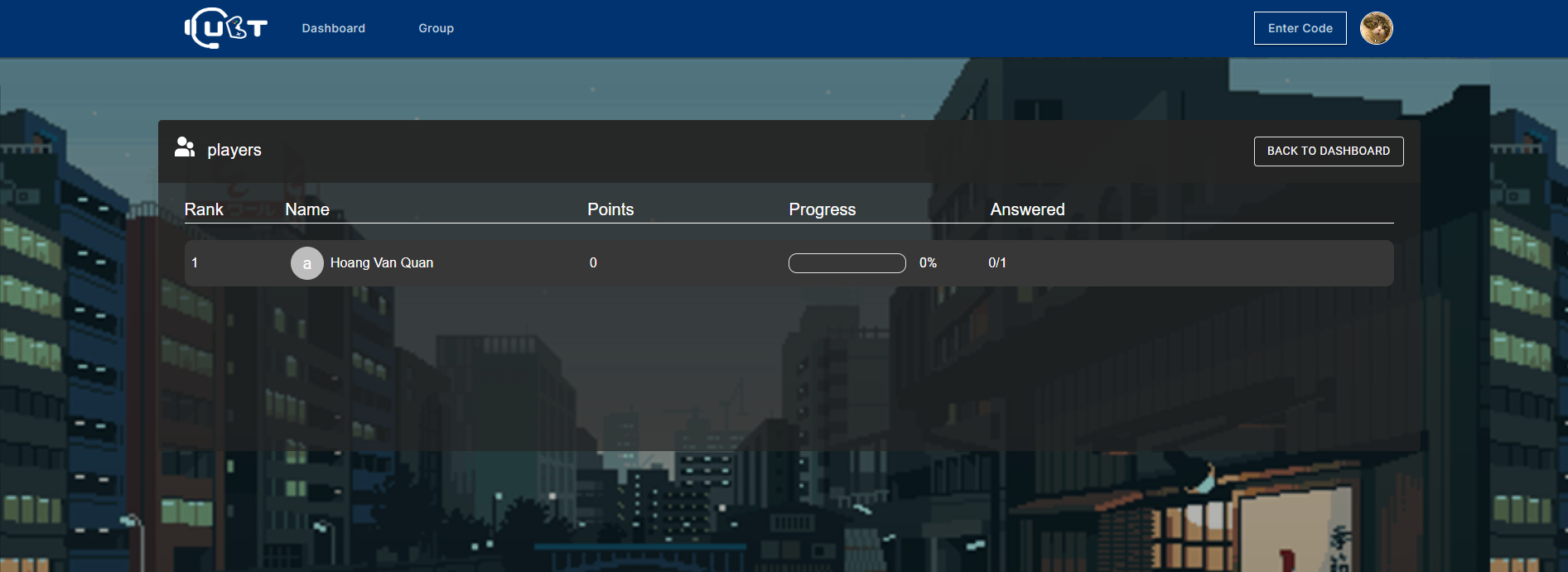
* Để tấn công quái vật, bạn cần trả lời các câu hỏi được đưa ra.
* Trả lời đúng: Bạn sẽ tấn công và làm giảm máu của quái vật.
* Trả lời sai: Quái vật sẽ tấn công và làm giảm máu của bạn.
* Quái vật hết máu: Bạn giết được quái vật và tiếp tục cuộc hành trình.
* Bạn hết máu: Nhân vật của bạn chết và trò chơi kết thúc.





**Xếp hạng điểm số**

* Điểm số cuối cùng sẽ được xếp theo bảng xếp hạng (ranking) với những người cùng phòng chơi.
* Bạn có thể xem thứ hạng của mình và so sánh với những người chơi khác để biết ai có điểm số cao nhất và tiến độ tốt nhất.



**Lưu ý**

* Quan sát số máu (Heart) của bạn và vị trí di chuyển để chiến đấu hiệu quả và tồn tại lâu dài.
* Thu thập và sử dụng các vật phẩm một cách thông minh để tối đa hóa khả năng sống sót và tiêu diệt quái vật.

*Chúc bạn chơi game vui vẻ và đạt được điểm số cao!*

# PHẦN V: FUTURE WORK

### **TÍCH HỢP CƠ SỞ DỮ LIỆU VECTOR CHO LANGCHAIN**

**Mục tiêu:** Ứng dụng vẫn còn chưa tối ưu về khả năng lưu trữ và truy xuất thông tin, các, khi đẩy bài lên (docx) thì hệ thống còn lâu và chưa có tính cá nhân hóa, vì thế nâng cao khả năng lưu trữ và truy xuất thông tin của Langchain là điều cấp thiết nhằm giúp ứng dụng phát triển hơn

**Một số hướng xử lý tiêu biểu:**

* **Tích hợp cơ sở dữ liệu vector:** Triển khai các hệ thống cơ sở dữ liệu vector tiên tiến như Milvus hoặc Faiss vào Langchain, cho phép lưu trữ và truy xuất hiệu quả các vectơ đại diện cho ngữ nghĩa của văn bản và mã nguồn trong trò chơi.
* **Phát triển API chuyên biệt:** Xây dựng bộ giao diện lập trình ứng dụng (API) mới, cung cấp cho các nhà phát triển trò chơi khả năng truy cập và thao tác dữ liệu vectơ một cách linh hoạt và hiệu quả. Hay nói cách khác là ta cung cấp một platform cho phép mọi người tạo thêm nhiều minigame không kém phần thú vị.
* **Tối ưu hóa truy vấn vector:** Ứng dụng các kỹ thuật tối ưu hóa truy vấn vector, đảm bảo khả năng đáp ứng thời gian thực, đáp ứng yêu cầu khắt khe về tốc độ của các ứng dụng trò chơi.

**Lợi ích:**

* **Lưu trữ và truy xuất hiệu quả:** Cho phép lưu trữ và truy xuất các mô hình ngôn ngữ lớn (LLM) và các vectơ biểu diễn ngữ nghĩa phức tạp một cách tối ưu.
* **Nâng cao hiệu suất:** Tăng tốc độ truy cập dữ liệu ngữ nghĩa, đáp ứng yêu cầu của các ứng dụng trò chơi đòi hỏi khả năng xử lý nhanh chóng.
* **Mở rộng tính năng:** Mở ra tiềm năng phát triển các tính năng mới dựa trên vectơ ngữ nghĩa, như hệ thống đề xuất nội dung cá nhân hóa và tạo ra các nhân vật trò chơi có độ chân thực cao hay tạo ra những bộ câu hỏi phù hợp với từng người chơi.

### **XÂY DỰNG ỨNG DỤNG DATA-INTENSIVE**

**Mục tiêu:** Phát triển các phương pháp mới, tối ưu hóa hiệu suất của Langchain trong việc xử lý khối lượng dữ liệu khổng lồ, đáp ứng yêu cầu của các ứng dụng trò chơi với hàng triệu người.

**Một số giải pháp khả thi:**

* **Nén dữ liệu:** Nghiên cứu và ứng dụng các thuật toán nén dữ liệu tiên tiến, giảm thiểu kích thước của mô hình ngôn ngữ và dữ liệu trò chơi.
* **Lấy mẫu và phân tán dữ liệu:** Triển khai các kỹ thuật lấy mẫu và phân tán dữ liệu, cho phép phân chia dữ liệu trò chơi trên nhiều máy chủ, tối ưu hóa khả năng xử lý.
* **Tối ưu hóa truy vấn:** Ứng dụng các kỹ thuật tối ưu hóa truy vấn dữ liệu, giảm thời gian tải và tăng cường độ mượt mà trong trải nghiệm chơi trò chơi.

**Lợi ích:**

* **Triển khai ứng dụng quy mô lớn:** Cho phép triển khai Langchain cho các ứng dụng trò chơi có quy mô lớn, phục vụ lượng người chơi đông đảo.
* **Giảm chi phí:** Giảm thiểu chi phí liên quan đến hạ tầng và nâng cao hiệu suất tổng thể của hệ thống.
* **Tăng cường trải nghiệm người dùng:** Mang đến trải nghiệm chơi trò chơi mượt mà, liền mạch và hấp dẫn hơn cho người dùng.

### **PHÁT TRIỂN HỆ THỐNG PHÂN TÁN**

**Mục tiêu:** Xây dựng một hệ thống phân tán có khả năng mở rộng linh hoạt, hỗ trợ đồng thời một lượng lớn người chơi truy cập vào trò chơi trong cùng một thời điểm.

**Giải pháp:**

* **Phân chia hệ thống:** Phân chia hệ thống Langchain thành nhiều máy chủ, phân tán tải trọng xử lý và tối ưu hóa hiệu suất.
* **Đồng bộ hóa dữ liệu:** Triển khai các cơ chế đồng bộ hóa dữ liệu giữa các máy chủ, đảm bảo tính nhất quán và toàn vẹn của dữ liệu.
* **Tối ưu hóa giao tiếp:** Phát triển các thuật toán giao tiếp hiệu quả, giảm thiểu độ trễ và tối ưu hóa hiệu suất mạng.

**Lợi ích:**

* **Tạo ra thế giới ảo lớn:** Cho phép xây dựng các thế giới trò chơi ảo rộng lớn, phức tạp và hỗ trợ nhiều người chơi đồng thời.
* **Tăng khả năng mở rộng:** Nâng cao khả năng mở rộng của hệ thống, đáp ứng nhu cầu ngày càng tăng của người dùng.
* **Nâng cao trải nghiệm trực tuyến:** Mang đến trải nghiệm chơi trò chơi trực tuyến mượt mà, hấp dẫn và thú vị hơn cho người chơi.

### **MỞ RỘNG VÀ CẢI TIẾN TRÒ CHƠI**

* **Nhiệm vụ và Cốt truyện**
* Thêm Cốt truyện: Tạo một cốt truyện hấp dẫn để người chơi cảm thấy có mục đích và động lực trong việc tiến hành cuộc hành trình.
* Nhiệm vụ Phụ: Thêm các nhiệm vụ phụ mà người chơi có thể hoàn thành để nhận phần thưởng như điểm kinh nghiệm, vật phẩm quý hiếm, hoặc tiền tệ trong game.
* **Vật phẩm và Trang bị**
* Vũ khí và Trang bị Đa dạng: Cung cấp nhiều loại vũ khí và trang bị khác nhau với các thuộc tính đặc biệt để người chơi có thể lựa chọn và tùy chỉnh nhân vật của mình.
* Nâng Cấp Vật Phẩm: Cho phép người chơi nâng cấp vũ khí và trang bị để tăng sức mạnh và khả năng chiến đấu.
* **Quái vật và Boss**
* Quái vật Đa dạng: Tạo ra nhiều loại quái vật với các kỹ năng và chiến thuật khác nhau để người chơi phải nghĩ cách đối phó.
* Trận Đấu Boss: Thêm các trận đấu boss định kỳ hoặc khi người chơi đạt đến một mốc nhất định, cung cấp thử thách lớn hơn và phần thưởng lớn.
* **Hệ thống Kỹ năng và Nâng cấp**
* Cây Kỹ năng: Tạo một cây kỹ năng để người chơi có thể chọn và nâng cấp kỹ năng theo ý muốn, từ đó tạo ra các chiến thuật chơi khác nhau.
* Cấp Độ Nhân Vật: Cho phép nhân vật lên cấp khi đạt đủ điểm kinh nghiệm, mở khóa kỹ năng mới và tăng cường chỉ số.
* **Thời gian và Sự kiện Đặc biệ**t
* Sự kiện Đặc biệt: Tổ chức các sự kiện trong game như lễ hội, cuộc thi hoặc sự kiện theo mùa để thu hút người chơi.
* Ngày và Đêm: Thêm cơ chế ngày và đêm để tạo thêm yếu tố chiến thuật, ví dụ như quái vật mạnh hơn vào ban đêm.
* **Chế độ Chơi Mới**
* Chế độ Sinh tồn: Tạo một chế độ chơi sinh tồn, nơi người chơi phải sống sót trong thời gian dài nhất có thể.
* Chế độ PvP: Thêm chế độ chơi đối kháng giữa các người chơi (PvP) để tăng tính cạnh tranh.
* **Hệ thống Xã hội và Hợp tác**
* Chơi Đội: Cho phép người chơi hợp tác với bạn bè để hoàn thành nhiệm vụ và đánh boss.
* Bảng Xếp hạng: Xây dựng bảng xếp hạng toàn cầu hoặc theo tuần để người chơi có thể so tài và đạt danh hiệu.
* **Tùy chỉnh Nhân vật**
* Trang phục và Giao diện: Cung cấp nhiều lựa chọn tùy chỉnh về trang phục và giao diện để người chơi có thể tạo ra nhân vật độc đáo của riêng mình.
* **Hệ thống Thưởng và Thành tựu**
* Thành Tựu: Tạo ra hệ thống thành tựu với các mục tiêu khác nhau để người chơi hoàn thành, từ đó nhận được phần thưởng đặc biệt.
* Hộp Quà Bí Ẩn: Thêm các hộp quà bí ẩn có thể tìm thấy trong trò chơi, chứa các vật phẩm ngẫu nhiên và có giá trị.
* **Giao diện và Trải nghiệm Người dùng**
* Giao diện Cải tiến: Nâng cấp giao diện người dùng để trực quan và dễ sử dụng hơn.
* Hướng dẫn và Tutorial: Cung cấp hướng dẫn chi tiết và tutorial để người chơi mới dễ dàng làm quen với game.

### **KẾT LUẬN**

Ba hướng phát triển chính trên là bên hướng đi giúp ứng dụng tiếp cận được nhiều người hơn cũng như đáp ứng được số lượng người chơi tăng đột biến cho tương lai. Ngoài ra, với cách tiếp cận của 3 hướng phát triển chính này bản thân ứng dụng cũng sẽ trở nên tối ưu hơn, tạo nên trải nghiệm tốt hơn cho người dùng.

# PHẦN VI: NGUỒN, TÀI LIỆU THAM KHẢO

### **LANGCHAIN**

* Trang chủ LangChain:<https://python.langchain.com/> - Trang web chính thức của LangChain, cung cấp thông tin tổng quan, hướng dẫn cài đặt và các ví dụ sử dụng cơ bản.
* Tài liệu LangChain: [Introduction | 🦜️🔗 LangChain](https://python.langchain.com/v0.1/docs/get_started/introduction/) Kho tài liệu chi tiết của LangChain, bao gồm hướng dẫn sử dụng, giải thích các khái niệm, API reference và các ví dụ thực tế giúp bạn tận dụng tối đa sức mạnh của LangChain.
* GitHub LangChain:<https://github.com/hwchase17/langchain> - Kho lưu trữ mã nguồn mở của LangChain trên GitHub, nơi bạn có thể tìm thấy mã nguồn, báo cáo lỗi, đóng góp vào dự án và thảo luận với cộng đồng nhà phát triển LangChain.

### **PHASER**

* Trang chủ Phaser:<https://phaser.io/> - Trang web chính thức của Phaser, cung cấp thông tin giới thiệu, tin tức, showcase các trò chơi được xây dựng bằng Phaser và liên kết đến các tài nguyên khác.
* Tài liệu Phaser:<https://photonstorm.github.io/phaser3-docs/index.html> - Tài liệu hướng dẫn chi tiết về Phaser, bao gồm các khái niệm cơ bản, hướng dẫn sử dụng từng thành phần, API reference và các ví dụ minh họa giúp bạn xây dựng trò chơi bằng Phaser.
* Ví dụ Phaser:<https://phaser.io/examples> - Bộ sưu tập các ví dụ trò chơi đa dạng được xây dựng bằng Phaser, từ các trò chơi đơn giản đến phức tạp, giúp bạn học hỏi và lấy cảm hứng cho dự án của mình.
* GitHub Phaser:<https://github.com/photonstorm/phaser> - Kho lưu trữ mã nguồn mở của Phaser trên GitHub, nơi bạn có thể tìm thấy mã nguồn, báo cáo lỗi, đóng góp vào dự án và thảo luận với cộng đồng nhà phát triển Phaser.

### **FIREBASE**

* Trang chủ Firebase:<https://firebase.google.com/> - Trang web chính thức của Firebase, giới thiệu các tính năng, sản phẩm và dịch vụ của Firebase, cũng như cung cấp các tài nguyên học tập và hỗ trợ.
* Tài liệu Firebase:<https://firebase.google.com/docs> - Trung tâm tài liệu của Firebase, bao gồm hướng dẫn sử dụng từng sản phẩm, API reference, các ví dụ mã nguồn và các bài viết hướng dẫn chi tiết giúp bạn tích hợp Firebase vào dự án của mình.
* Blog Firebase:<https://firebase.googleblog.com/> - Blog chính thức của Firebase, cung cấp thông tin về các bản cập nhật mới, các mẹo và thủ thuật sử dụng Firebase, cũng như các câu chuyện thành công từ cộng đồng nhà phát triển Firebase.

### **NEXT.JS**

* Trang chủ Next.js:<https://nextjs.org/> - Trang web chính thức của Next.js, giới thiệu về Next.js, các tính năng nổi bật, các trường hợp sử dụng và các tài nguyên học tập.
* Tài liệu Next.js:<https://nextjs.org/docs> - Tài liệu hướng dẫn chi tiết về Next.js, bao gồm các khái niệm cơ bản, hướng dẫn sử dụng từng tính năng, API reference và các ví dụ minh họa giúp bạn xây dựng ứng dụng web bằng Next.js.
* Học Next.js:<https://nextjs.org/learn> - Các khóa học và hướng dẫn trực tuyến giúp bạn học Next.js từ cơ bản đến nâng cao, bao gồm các bài tập thực hành và dự án mẫu.
* GitHub Next.js: [đã xoá URL không hợp lệ] - Kho lưu trữ mã nguồn mở của Next.js trên GitHub, nơi bạn có thể tìm thấy mã nguồn, báo cáo lỗi, đóng góp vào dự án và thảo luận với cộng đồng nhà phát triển Next.js.

### **LLM (LARGE LANGUAGE MODEL)**

* Bài viết tổng quan về LLM:<https://en.wikipedia.org/wiki/Large_language_model> - Bài viết trên Wikipedia cung cấp thông tin tổng quan về LLM, bao gồm định nghĩa, lịch sử phát triển, các kiến trúc phổ biến, ứng dụng và các vấn đề liên quan.
* Khóa học về LLM (Coursera):<https://www.coursera.org/specializations/natural-language-processing> - Chuyên ngành về xử lý ngôn ngữ tự nhiên trên Coursera, bao gồm các khóa học về LLM, Transformers, và các kỹ thuật xử lý ngôn ngữ tự nhiên khác.
* Sách về LLM:
  + "Natural Language Processing with Transformers" by Lewis Tunstall, Leandro von Werra, Thomas Wolf - Sách cung cấp kiến thức toàn diện về Transformers và cách sử dụng chúng trong xử lý ngôn ngữ tự nhiên, bao gồm các ứng dụng như LLM.
  + "Deep Learning for Natural Language Processing" by Yoav Goldberg - Sách giới thiệu về các phương pháp học sâu trong xử lý ngôn ngữ tự nhiên, bao gồm các kiến trúc mạng nơ-ron và các ứng dụng thực tế.

### **PROMPT ENGINEERING**

* Learn Prompting:<https://learnprompting.org/> - Trang web cung cấp các hướng dẫn, bài viết và tài nguyên về prompt engineering, giúp bạn hiểu và áp dụng các kỹ thuật để tương tác hiệu quả với các mô hình ngôn ngữ lớn.
* Prompt Engineering Guide:<https://www.promptingguide.ai/> - Hướng dẫn chi tiết về prompt engineering, bao gồm các khái niệm cơ bản, các kỹ thuật nâng cao và các ví dụ thực tế giúp bạn tối ưu hóa hiệu suất của mô hình ngôn ngữ trong các tác vụ cụ thể.
* The Art of Prompt Engineering (free ebook):<https://www.promptingguide.ai/> - Sách điện tử miễn phí cung cấp kiến thức và kỹ thuật về prompt engineering, giúp bạn hiểu cách thức hoạt động của mô hình ngôn ngữ và cách xây dựng các prompt hiệu quả.