

**TÀI LIỆU CHỨC NĂNG CƠ BẢN**

**Chức năng cơ bản của dự án**

**Được soạn bởi: Nguyễn Thị Lan Hương**

**Nguyễn Minh Dũng**

**Chu Quang Cần**

**Lại Hoàng Hiệp**

**Trần Huy Hoàng**

**Trường Đại học Công nghệ, ĐHQGHN**

**Hà Nội, 26-03-2024**

**Mục lục**

[**Lịch sử sửa đổi 3**](#_Toc165841299)

[**1. Giới thiệu 4**](#_Toc165841300)

[1.1. Mục đích sản phẩm 4](#_Toc165841301)

[1.2. Chú thích các thuật ngữ 5](#_Toc165841302)

[1.3. Phạm vi 6](#_Toc165841303)

[**2. Demo các chức năng chính 6**](#_Toc165841304)

[2.1. Đăng ký / Đăng nhập: 6](#_Toc165841305)

[2.2. Flashcard: 8](#_Toc165841306)

[2.3. Forum: 14](#_Toc165841307)

[2.4. Games: 17](#_Toc165841308)

[**3. Cách hoạt động của các chức năng chính 19**](#_Toc165841309)

[3.1. Đăng nhập / đăng ký: 19](#_Toc165841310)

[3.2. Flashcard: 20](#_Toc165841311)

[3.3. Forum: 21](#_Toc165841312)

[3.4. Game: 22](#_Toc165841313)

[**4. Thách thức và đề xuất giải pháp 23**](#_Toc165841314)

[4.1. Thách thức / Giải pháp về thời gian phát triển 23](#_Toc165841315)

[4.2. Thách thức / Giải pháp về thực hiện đúng yêu cầu dự án 24](#_Toc165841316)

[4.3. Thách thức / Giải pháp về bảo mật 24](#_Toc165841317)

[4.4. Thách thức / Giải pháp liên quan đến nội dung bẩn 24](#_Toc165841318)

# Lịch sử sửa đổi

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ và tên** | **Thời gian** | **Lý do sửa đổi** | **Phiên bản** |
| Lại Hoàng Hiệp | 26/04/2024 | Khởi tạo mẫu tài liệu và giới thiệu sản phẩm | 1.0 |
| Lại Hoàng Hiệp | 30/04/2024 | Hoàn hiện cách hoạt động và thách thức / giải pháp | 1.1 |
| Lại Hoàng Hiệp | 04/05/2024 | Hoàn thiện phần demo sản phẩm | 1.2 |

# Danh mục hình ảnh và biểu đồ

[1 - Giao diện đăng ký 6](#_Toc165841561)

[2 - Giao diện đăng nhập 7](#_Toc165841562)

[3 - Giao diện flashcard 8](#_Toc165841563)

[4 - Giao diện môn học 8](#_Toc165841564)

[5 - Giao diện mặt trước flashcard 9](#_Toc165841565)

[6 - Giao diện mặt sau flashcard 9](#_Toc165841566)

[7 - Giao diện nút thêm flashcard 10](#_Toc165841567)

[8 - Giao diện thêm flashcard 10](#_Toc165841568)

[9 - Giao diện thêm bộ flashcard 11](#_Toc165841569)

[10 - Giao diện thêm flashcard vào một bộ 11](#_Toc165841570)

[11 - Giao diện nút xóa bộ flashcard 12](#_Toc165841571)

[12 - Giao diện xóa bộ flashcard 12](#_Toc165841572)

[13 - Giao diện nút xóa flashcard 13](#_Toc165841573)

[14 - Giao diện xóa flashcard 14](#_Toc165841574)

[15 - Giao diện nút diễn đàn 14](#_Toc165841575)

[16 - Giao diện thêm bài viết 15](#_Toc165841576)

[17 - Giao diện tương tác diễn đàn 16](#_Toc165841577)

[18 - Giao diện nút trò chơi 16](#_Toc165841578)

[19 - Giao diện game 17](#_Toc165841579)

[20 - Giao diện tạo phòng 17](#_Toc165841580)

[21 - Giao diện gia nhập phòng 18](#_Toc165841581)

# Giới thiệu

## Mục đích sản phẩm

Ngày nay, internet ngày càng phát triển nhanh chóng, mọi hoạt động đều được diễn ra trên không gian mạng. Và việc học tập cũng không phải là ngoại lệ. Nhu cầu của học sinh, sinh viên ngày nay là mong muốn việc học tập được diễn ra ở bất kì đâu, bất kỳ lúc nào một cách nhanh chóng và hiệu quả. Vì vậy giải pháp học tập online trên các thiết bị điện tử, được cho là mang lại sự tiện lợi và hiệu quả học tập cao.

Nắm bắt được nhu cầu đó, chúng tôi quyết định xây dựng một hệ thống web app có tên là Quyts, hỗ trợ người học giúp đem lại giải pháp cho vấn đề trên. Chúng tôi hướng đến xây dựng một môi trường học tập tiện lợi, nhanh chóng và hiệu quả nhất, từ đó khuyến khích mọi người học tập và phát triển bản thân.

Quyts sẽ giúp người dùng được trải nghiệm nhiều phương pháp tiếp thu kiến thức một cách hiệu quả: flashcard, câu hỏi trắc nghiệm, rất nhiều game học tập thú vị khác nữa. Và hơn thế nữa, chúng tôi còn cung cấp một diễn đàn trao đổi kiến thức về tất cả lĩnh vực kiến thức, giúp người học giải đáp các thắc mắc và học tập từ mọi người xung quanh.

Đối tượng hướng đến của hệ thống hỗ trợ học tập này của chúng tôi là bất kể người dùng nào có mong muốn nâng cao hiệu suất học tập, đảm bảo tính linh hoạt về thời gian.

## Chú thích các thuật ngữ

Web của chúng tôi có thể có những thuật ngữ gây khó hiểu cho người đọc/dùng. Do đó, đây sẽ là phần làm sáng tỏ để người dùng có thể dễ dàng sử dụng:

* **Flashcard**: Là các thẻ học tập, gồm 2 mặt. Một mặt là thuật ngữ, một mặt là ý nghĩa/khái niệm.
* **Rank**: Xếp hạng mức độ “tích cực” của người dùng trên web. Khi một người up tài liệu lên và tài liệu đó được đánh dấu là hữu ích, người đó sẽ được tăng hạng. Hoặc khi giải đáp thắc mắc trên diễn đàn và câu trả lời đó đánh dấu là hữu ích, cũng sẽ được tăng hạng. Rank có 4 mức được xếp hạng từ sắt -> đồng -> bạc -> vàng -> kim cương.
* **Forum**: Là diễn đàn, nơi người dùng có thể trao đổi kiến thức với nhau
* **Client:** Phía người dùng
* **Server:** Phía máy chủ
* **Request:** Yêu cầu được gửi lên từ người dùng đến máy chủ
* **Response:** Phản hồi được máy chủ gửi đến người dùng

## Phạm vi

Website của chúng tôi hoạt động đồng thời trên nhiều thiết bị máy tính/điện thoại có kết nối mạng khác nhau bằng cách chia sẻ chung một cơ sở dữ liệu thay đổi trong thời gian thực.

Web của chúng tôi cung cấp nơi lưu trữ các tài liệu học tập và biến nó thành các hình thức học tập ghi nhớ thú vị hơn, hiệu quả hơn và tiện lợi hơn.

Website được phát triển để đáp ứng tính năng cơ bản trong khoảng thời gian là 3 tháng.

# Demo các chức năng chính

## Đăng ký / Đăng nhập:

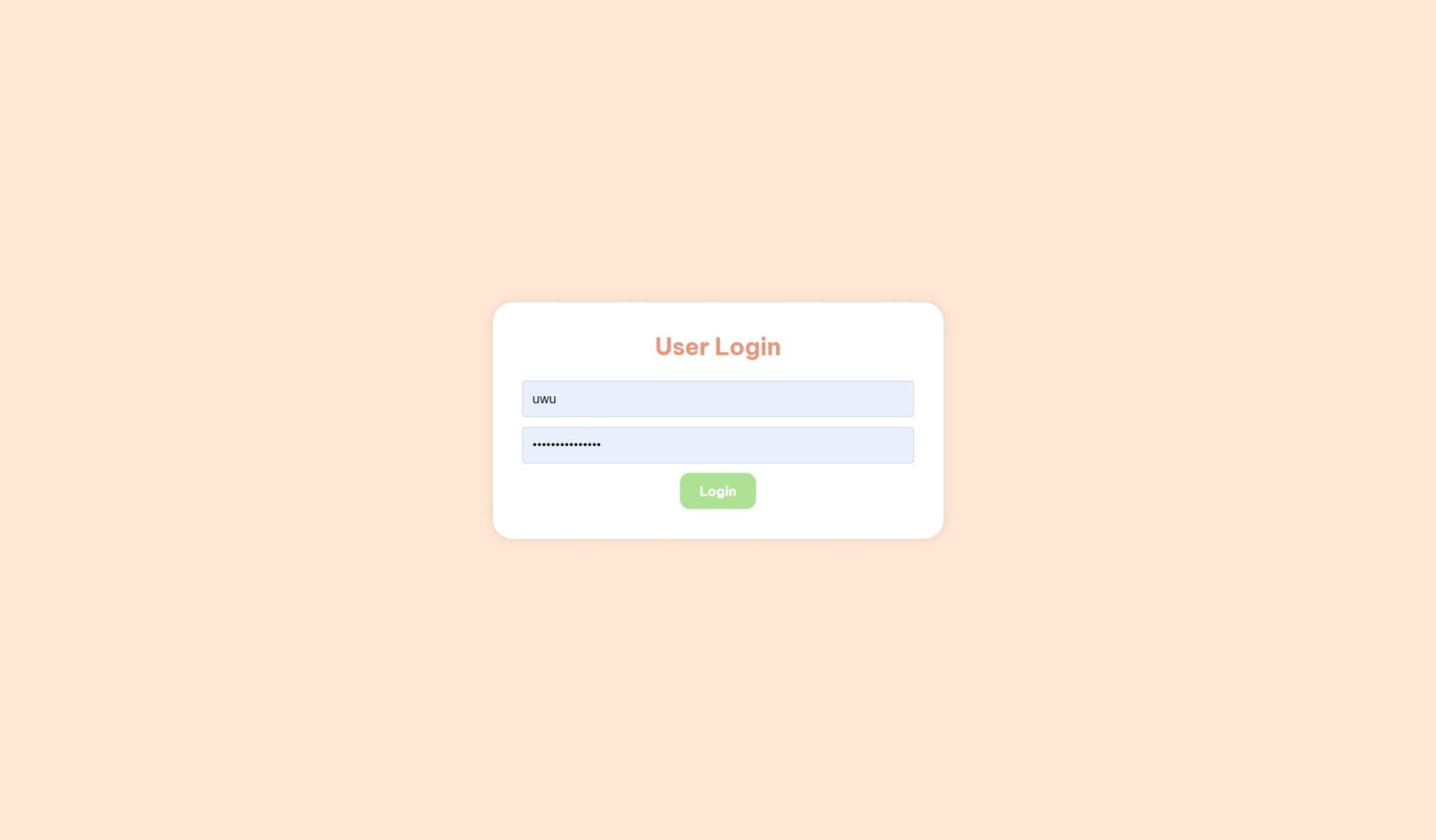
Để có thể sử dụng toàn bộ các chức năng chính của Web, người dùng cần phải có tài khoản để đăng nhập, hoặc đăng ký (nếu chưa có tài khoản)

* **Đối với đăng ký**: Người dùng sẽ cần điền đầy đủ thông tin vào 3 trường: Username (tên để hiển thị trong suốt quá trình dùng), Email (địa chỉ email để lấy lại mật khẩu nếu quên), Password.

Giao điện đăng ký tài khoản


1 - Giao diện đăng ký

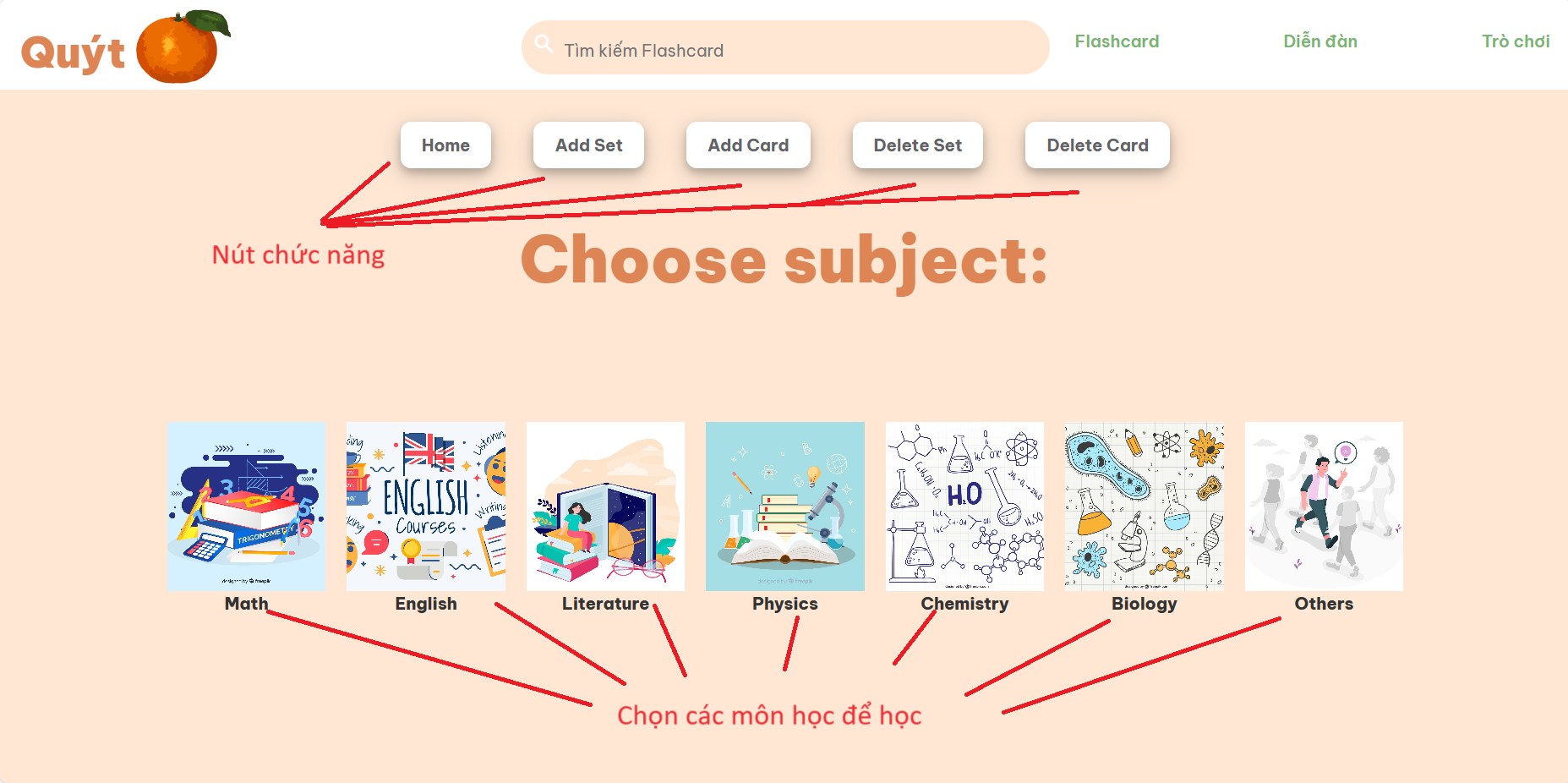
* **Đối với đăng nhập**: Sau khi đã tạo tài khoản thành công hoặc đã có tài khoản trước đó, người dùng cần đăng nhập để sử dụng mọi tính năng của Web. Người dùng cần điền username và mật khẩu của tài khoản đó.



2 - Giao diện đăng nhập

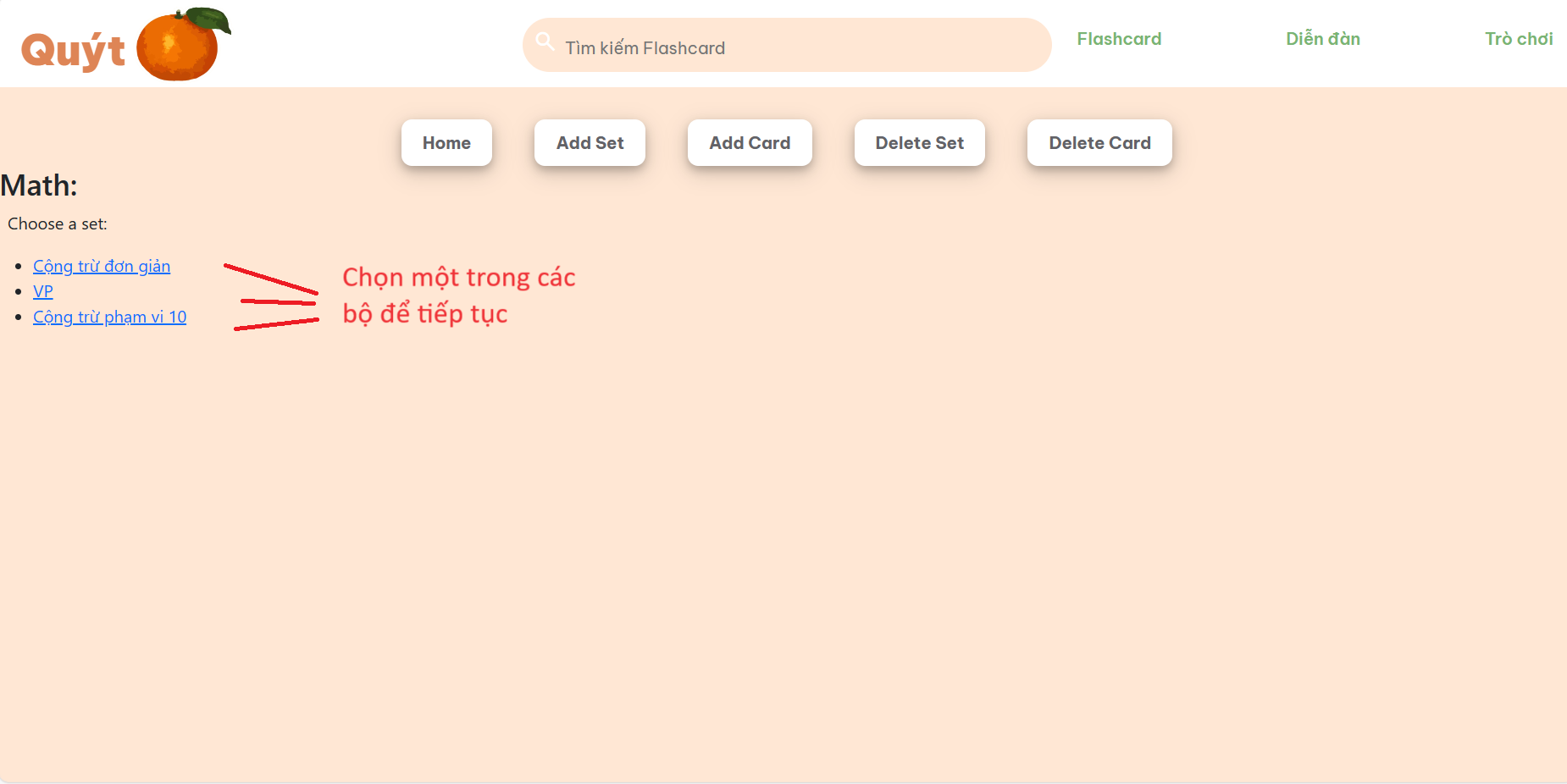
## Flashcard:

Sau khi đăng nhập thành công, trang chủ của Web sẽ hiển thị với nội dung được mô tả như bên dưới cho người dùng. Người dùng sau đó có thể chọn thêm/xóa Flashcard hoặc chọn một môn học để học



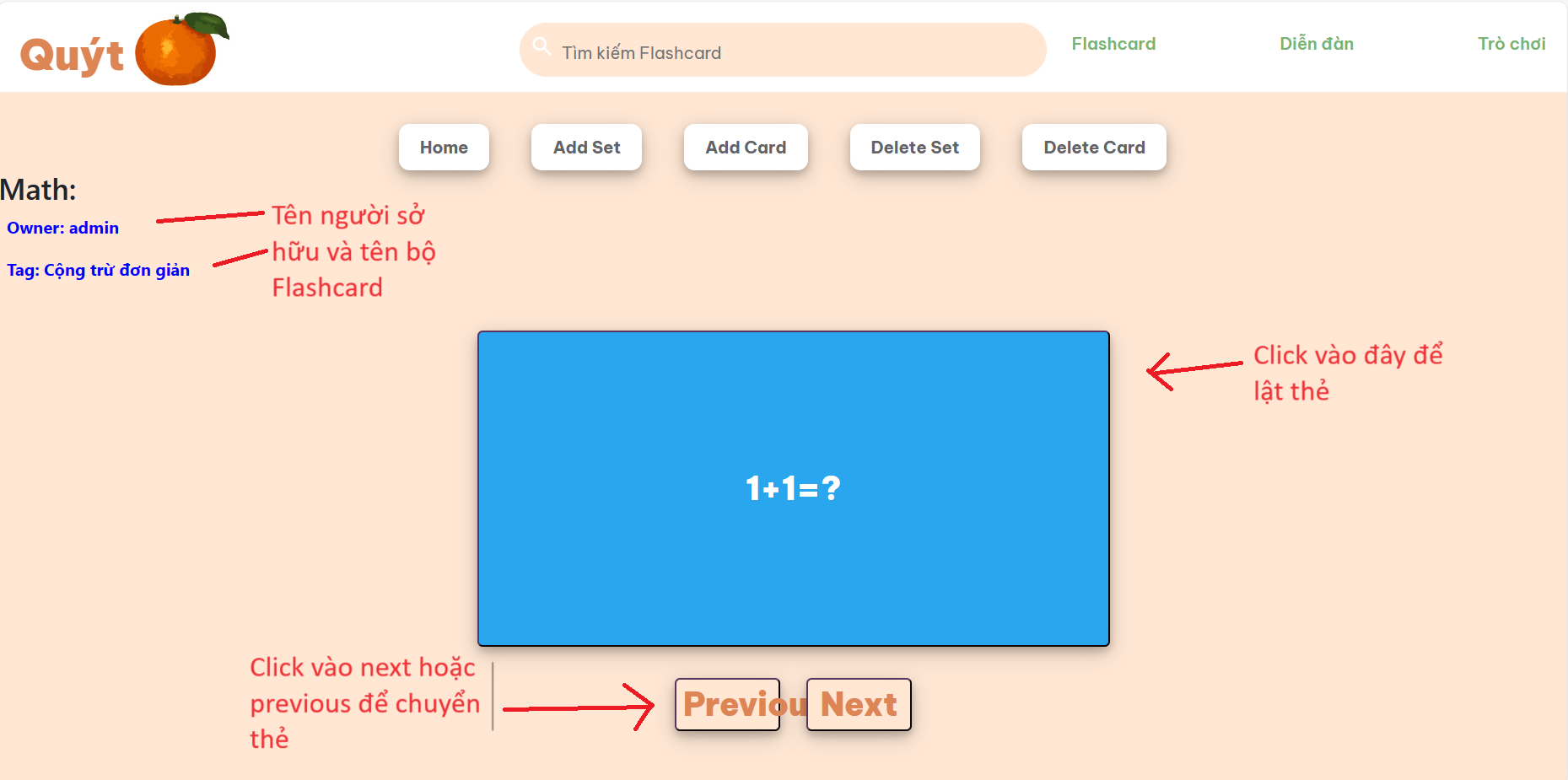
3 - Giao diện flashcard

* **Chọn môn học để học:** 
  + Nếu người dùng click vào một trong những môn học được hiển thị trên Web, giao diện như bên dưới sẽ hiện ra bao gồm các bộ/gói Flashcard của môn học đó. Ví dụ như click vào Math thì sẽ có giao diện như sau

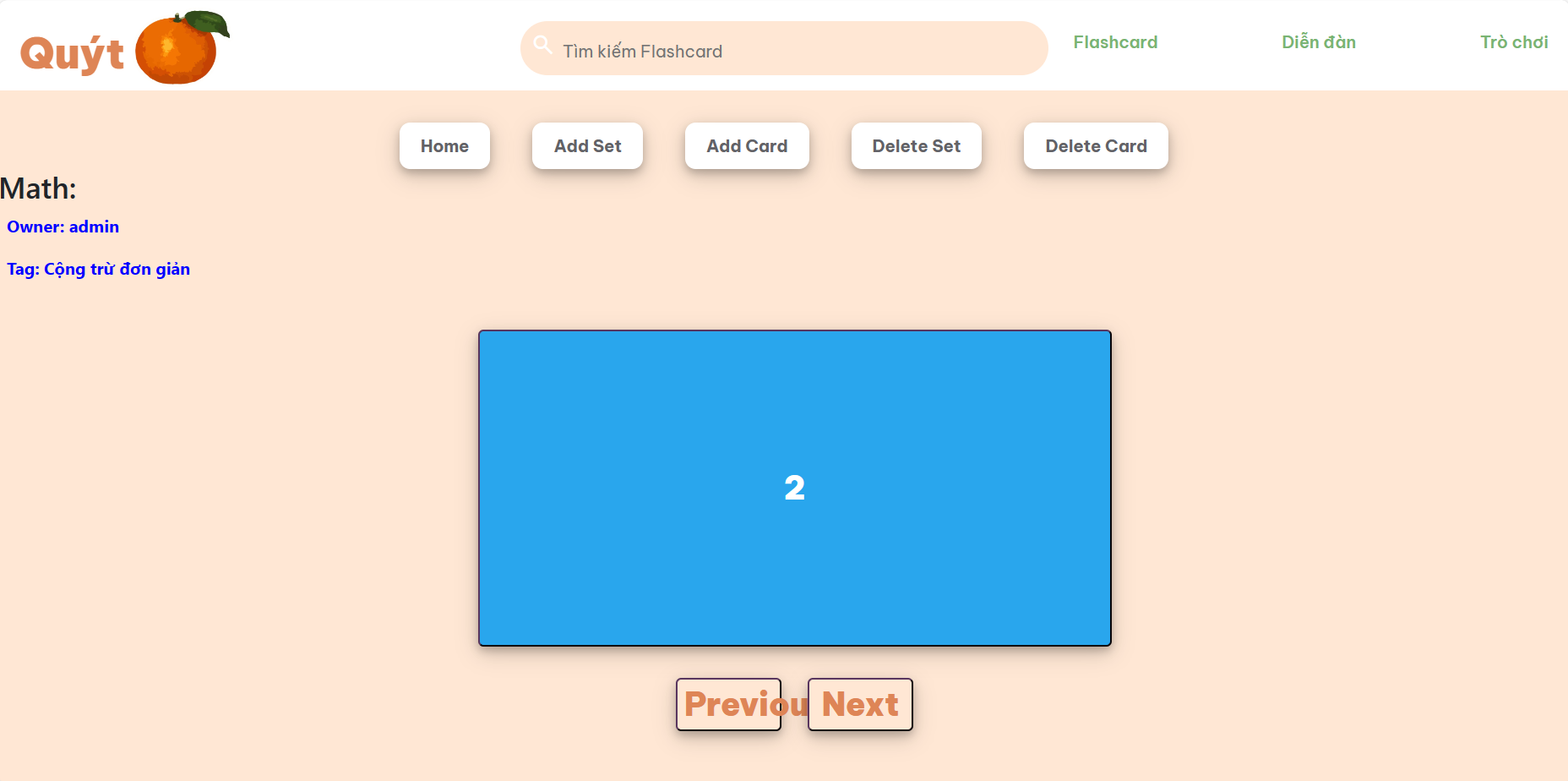


4 - Giao diện môn học

* + Sau khi click chọn một bộ/gói Flashcard, người dùng sẽ có thể bắt đầu học và thao tác cùng với Flashcard như bên dưới

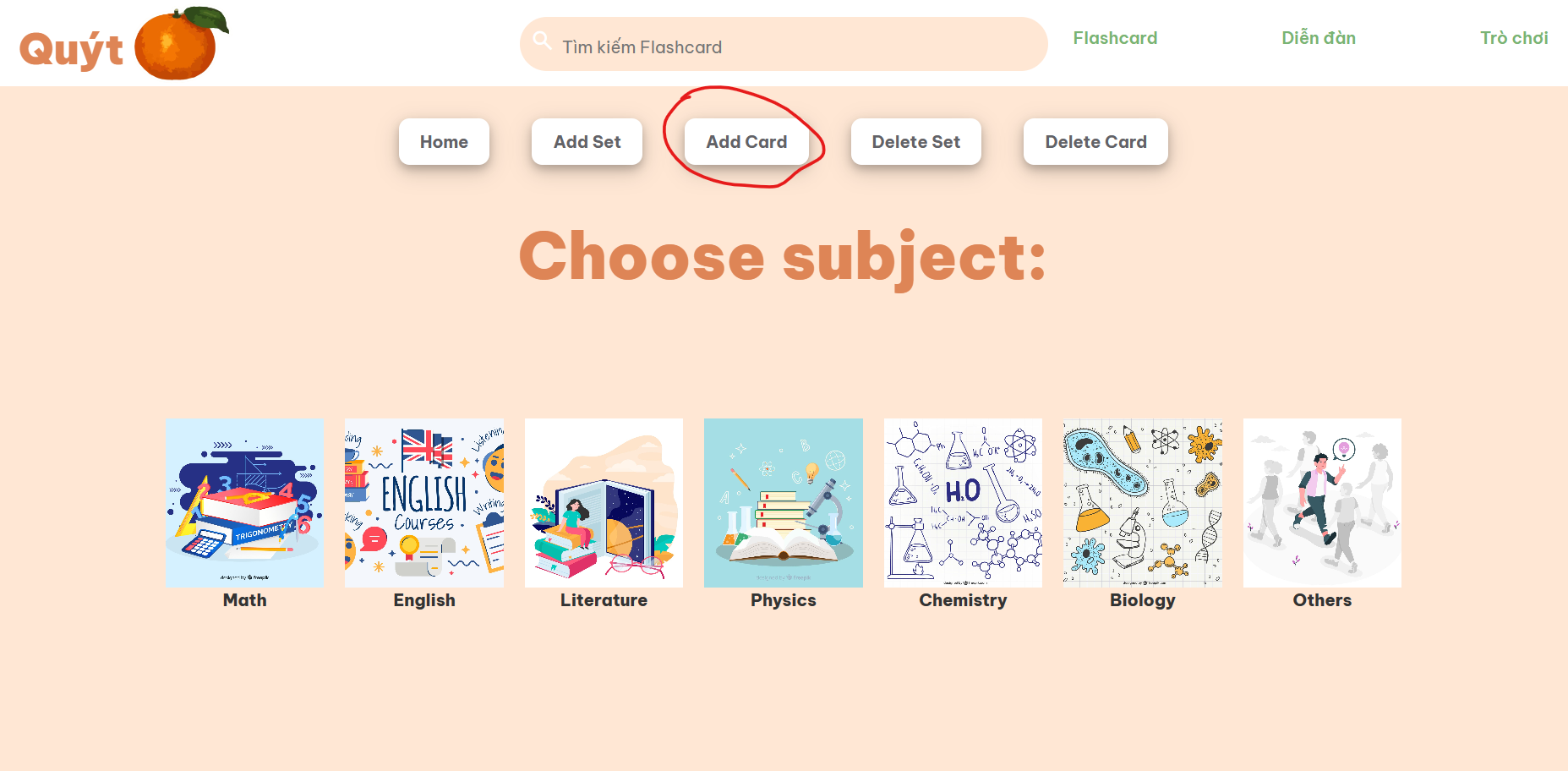


5 - Giao diện mặt trước flashcard

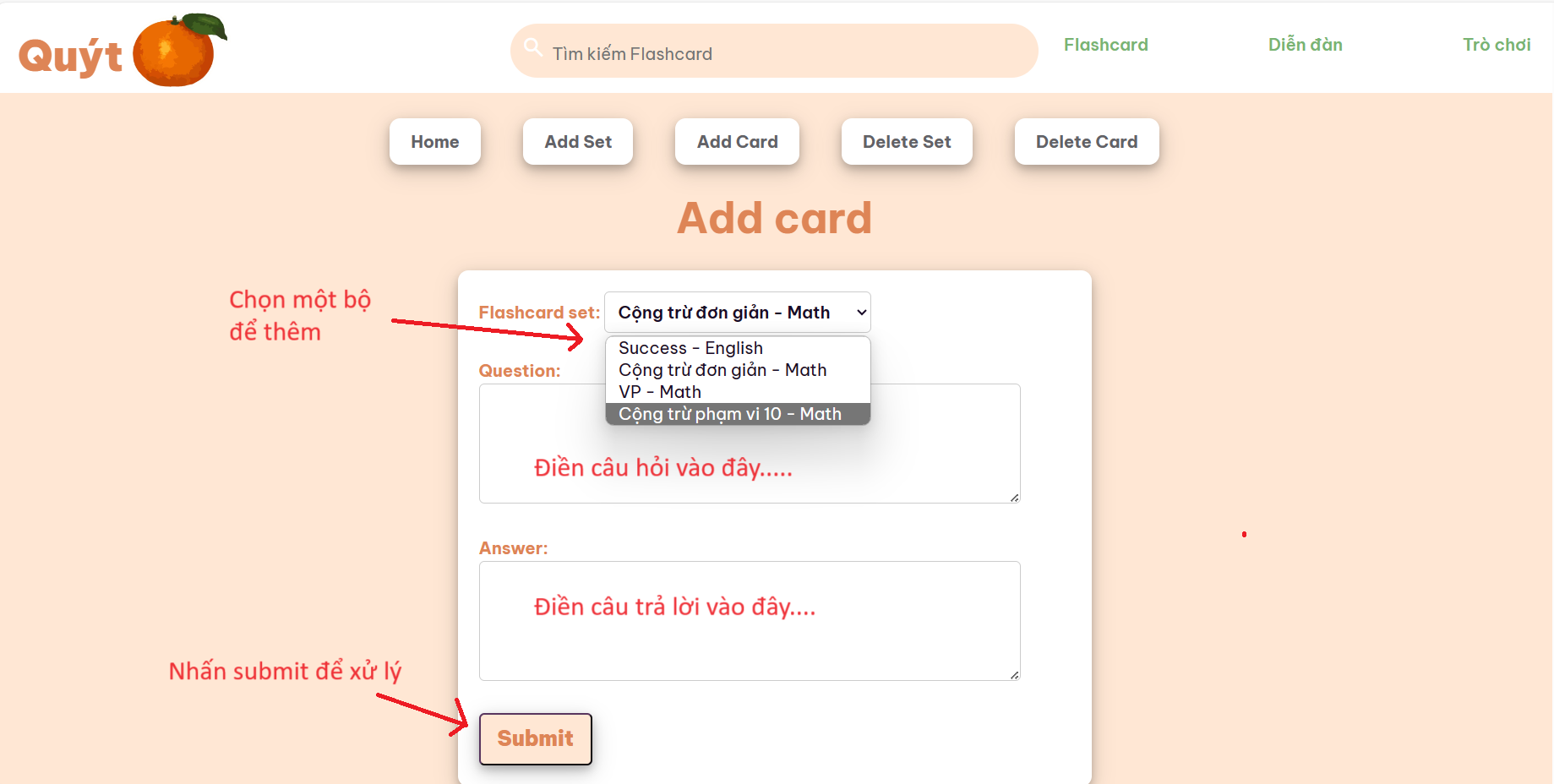


6 - Giao diện mặt sau flashcard

* **Thêm một Flashcard vào một bộ có sẵn:**
  + Người dùng click vào nút “Add Card” (luôn được hiển thị ở bên trên trong các nút chức năng) và một giao diện để thêm một Flashcard sẽ hiển thị như sau:

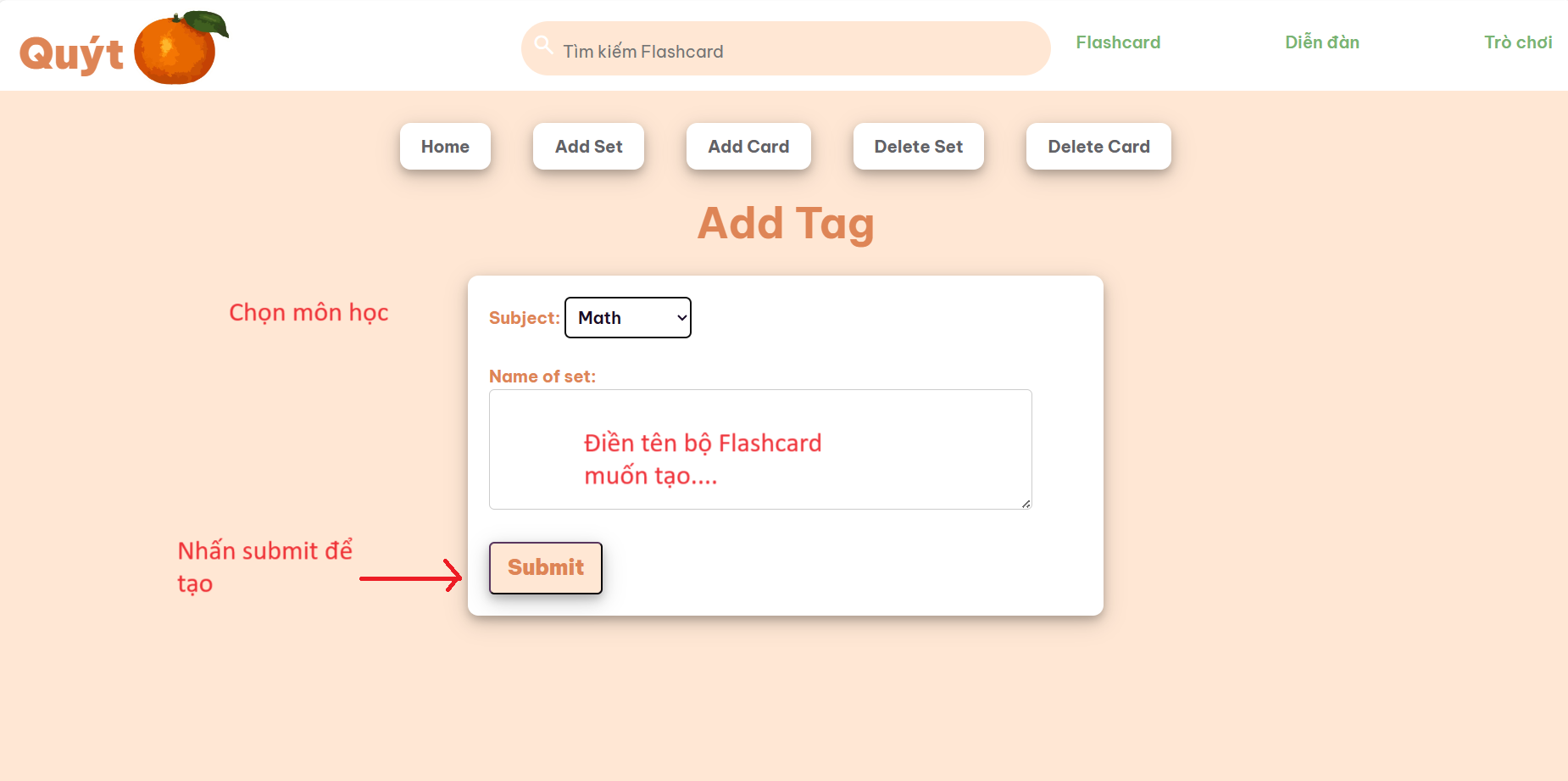


7 - Giao diện nút thêm flashcard

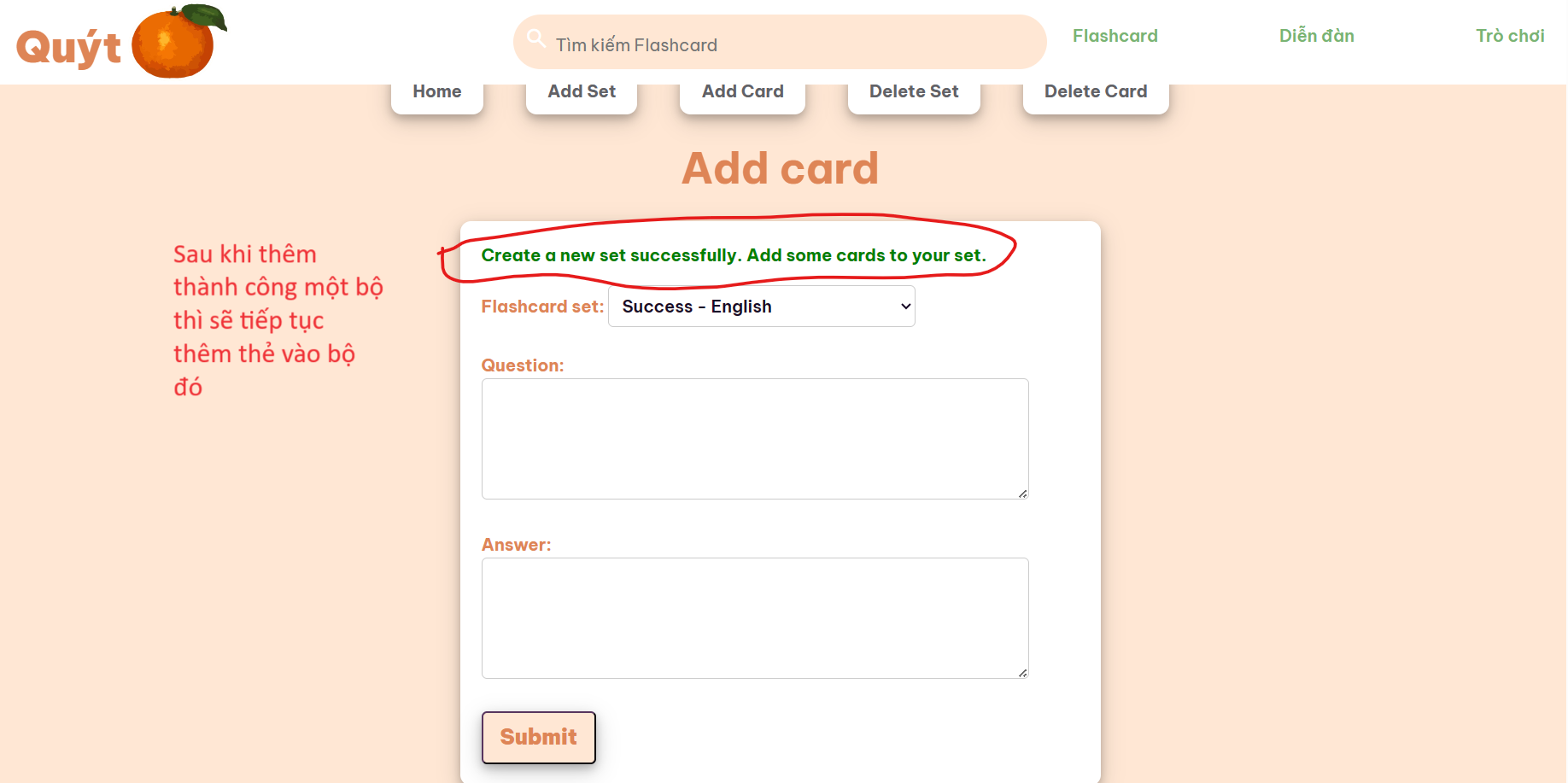


8 - Giao diện thêm flashcard

* **Thêm một bộ Flashcard:**
  + Người dùng click vào “Add Set”. Sau khi thêm thành công một bộ thì sẽ nhận được thông báo và chuyển đến trang để thêm từng Flashcard vào bộ đó

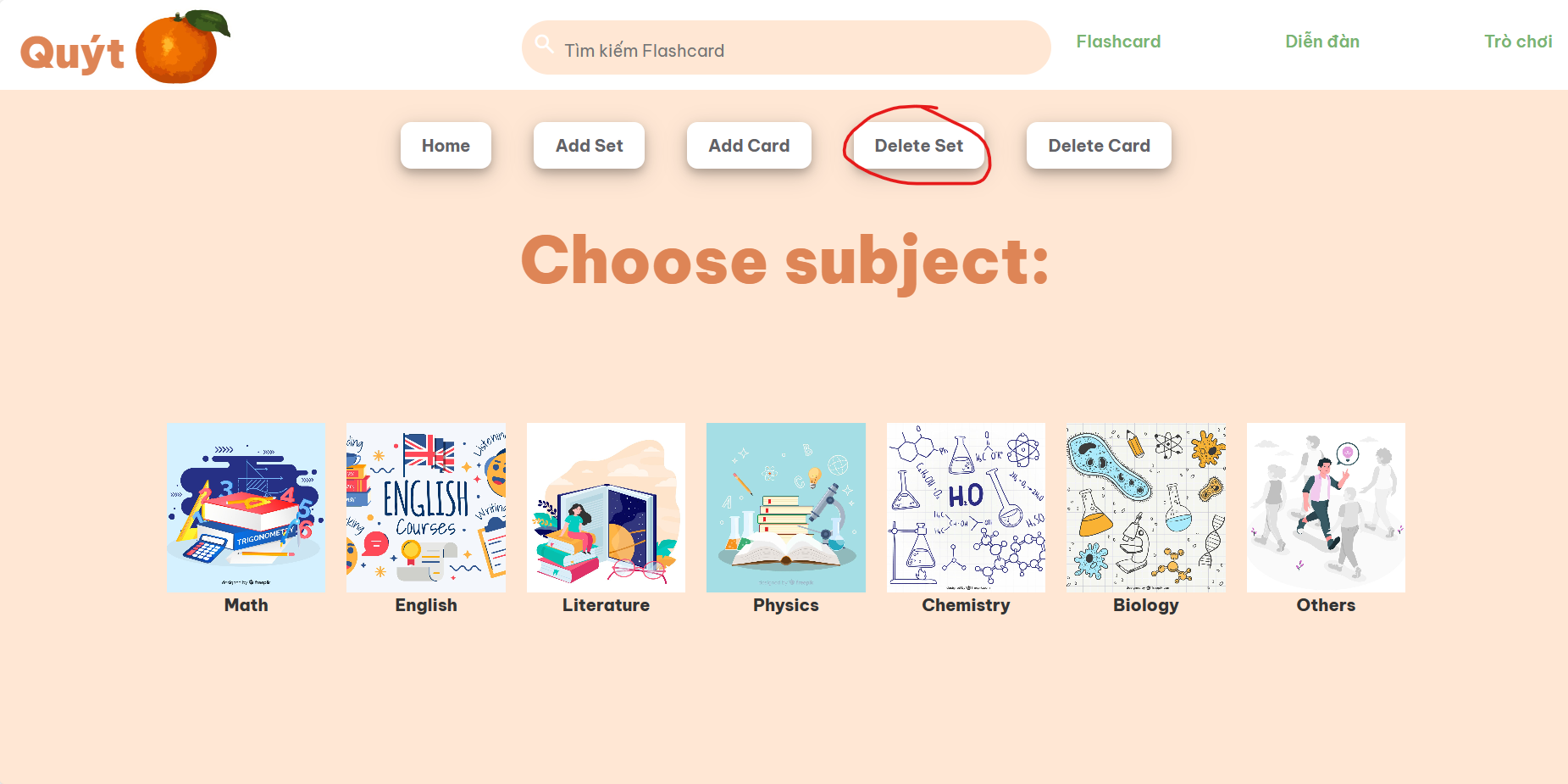


9 - Giao diện thêm bộ flashcard

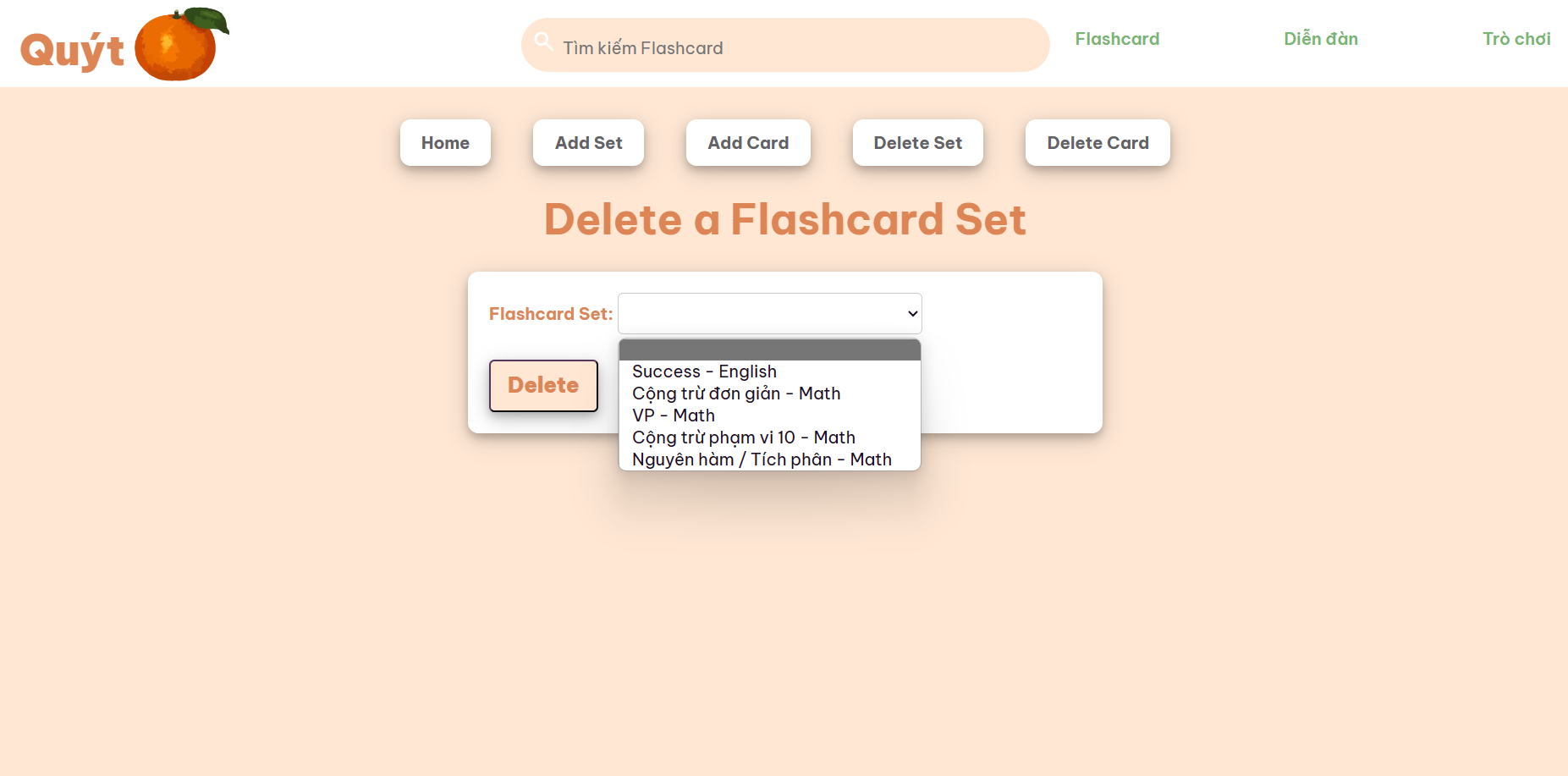


10 - Giao diện thêm flashcard vào một bộ

* **Xóa một bộ Flashcard**:
  + Người dùng click vào “Delete Set” để thực hiện xóa một bộ Flashcard do mình sở hữu (chỉ có thể xóa bộ Flashcard do mình tạo). Sau đó, người dùng sẽ tiến hành chọn bộ Flashcard muốn xóa và nhấn Delete.

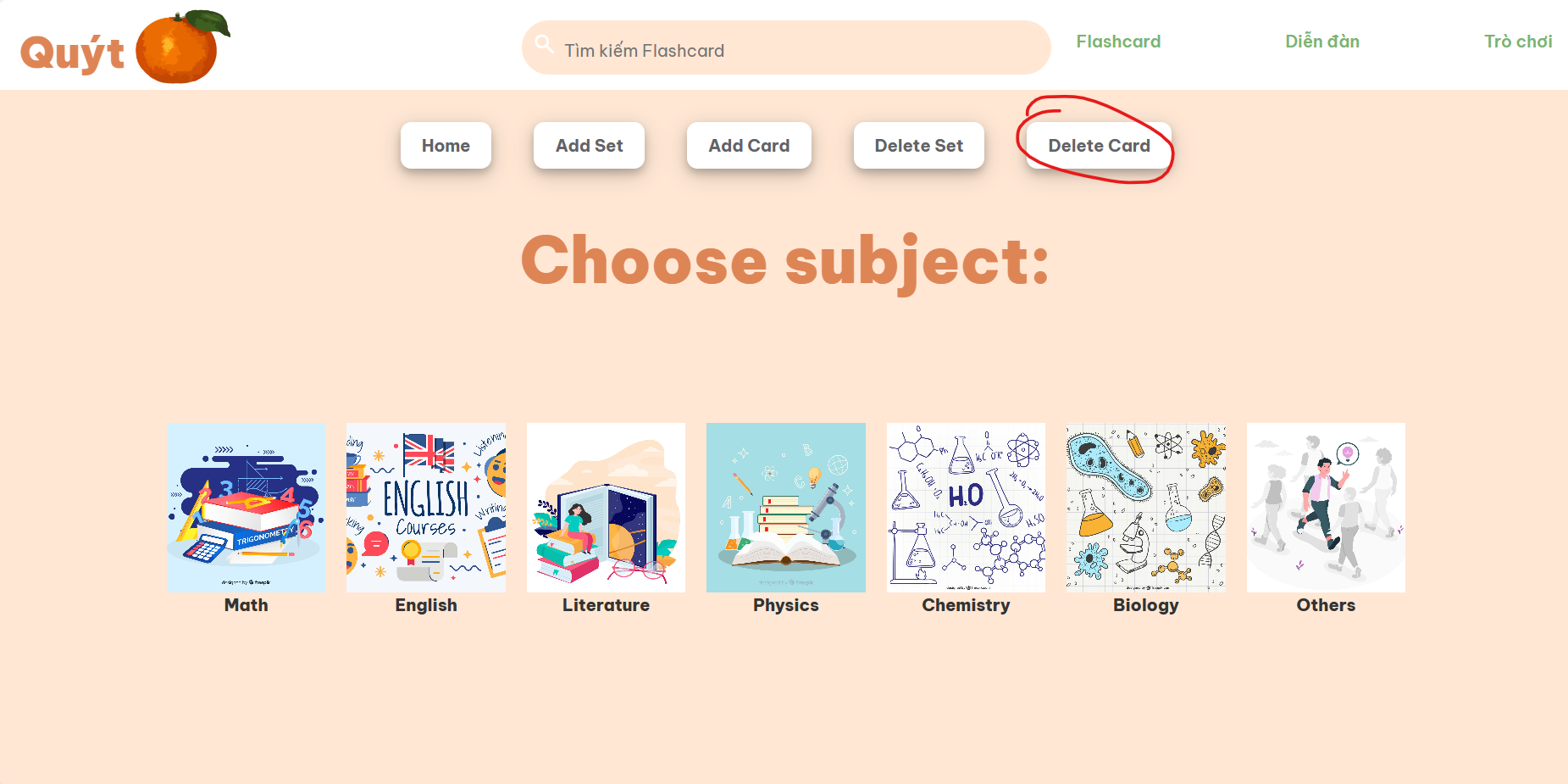


11 - Giao diện nút xóa bộ flashcard

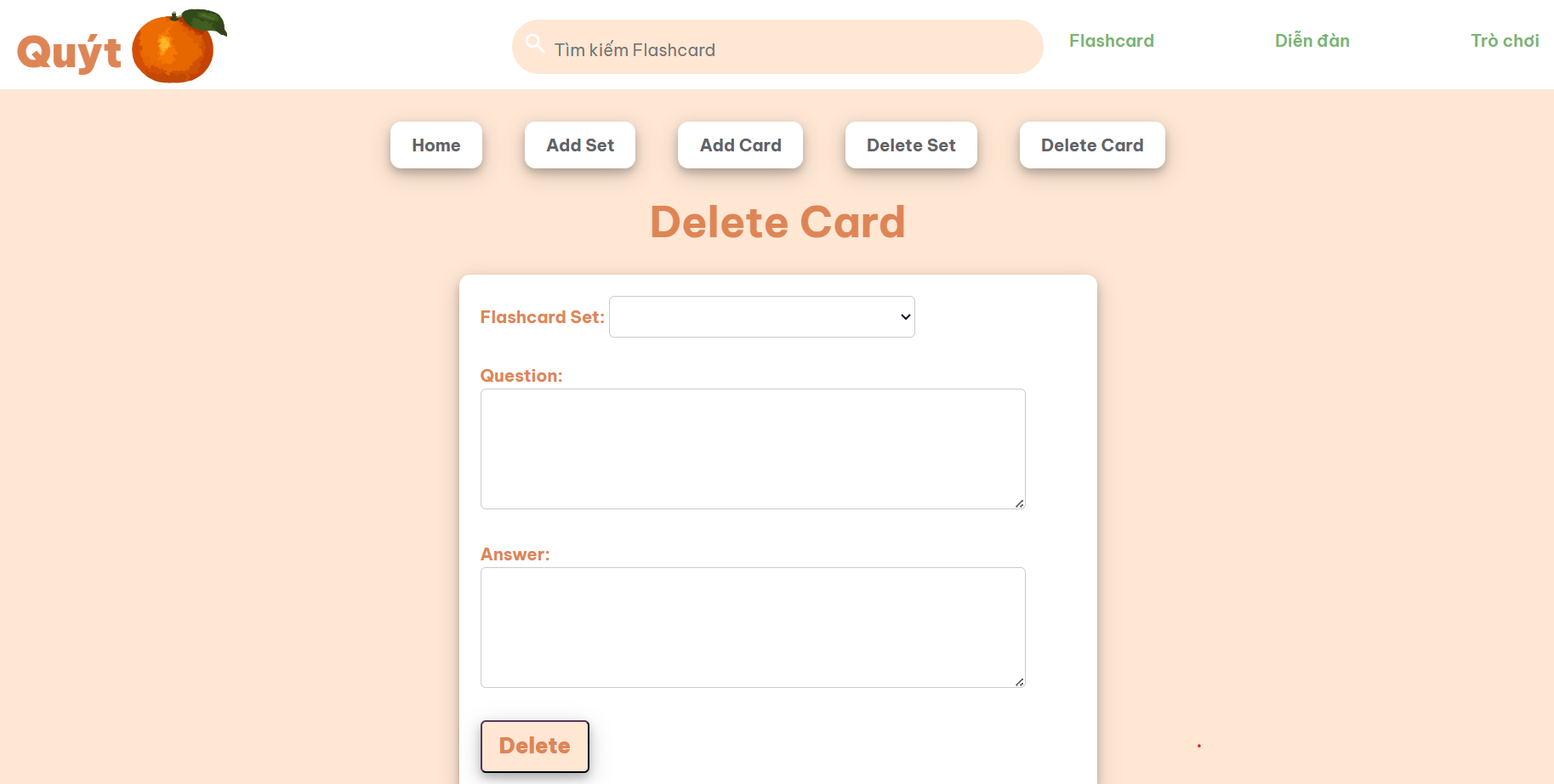


12 - Giao diện xóa bộ flashcard

* **Xóa một Flashcard**:
  + Người dùng click vào “Delete Card” để thực hiện xóa một bộ Flashcard do mình sở hữu (chỉ có thể xóa bộ Flashcard do mình tạo). Người dùng sẽ chọn bộ Flashcard của mình chứa Flashcard cần xóa đó, và chọn câu hỏi và câu trả lời để đảm bảo rằng người dùng thực sự muốn xóa Flashcard đó (giao diện cũng giống như thêm một thẻ).



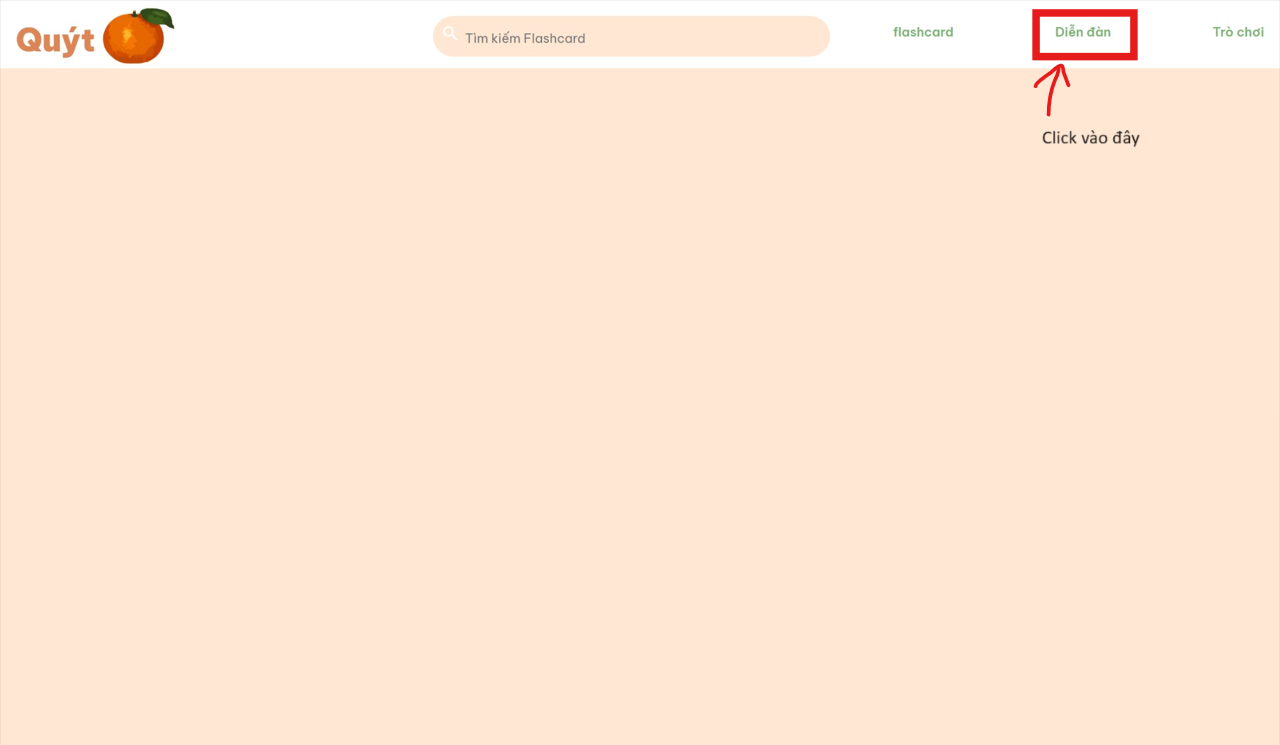
13 - Giao diện nút xóa flashcard



14 - Giao diện xóa flashcard

## Forum:

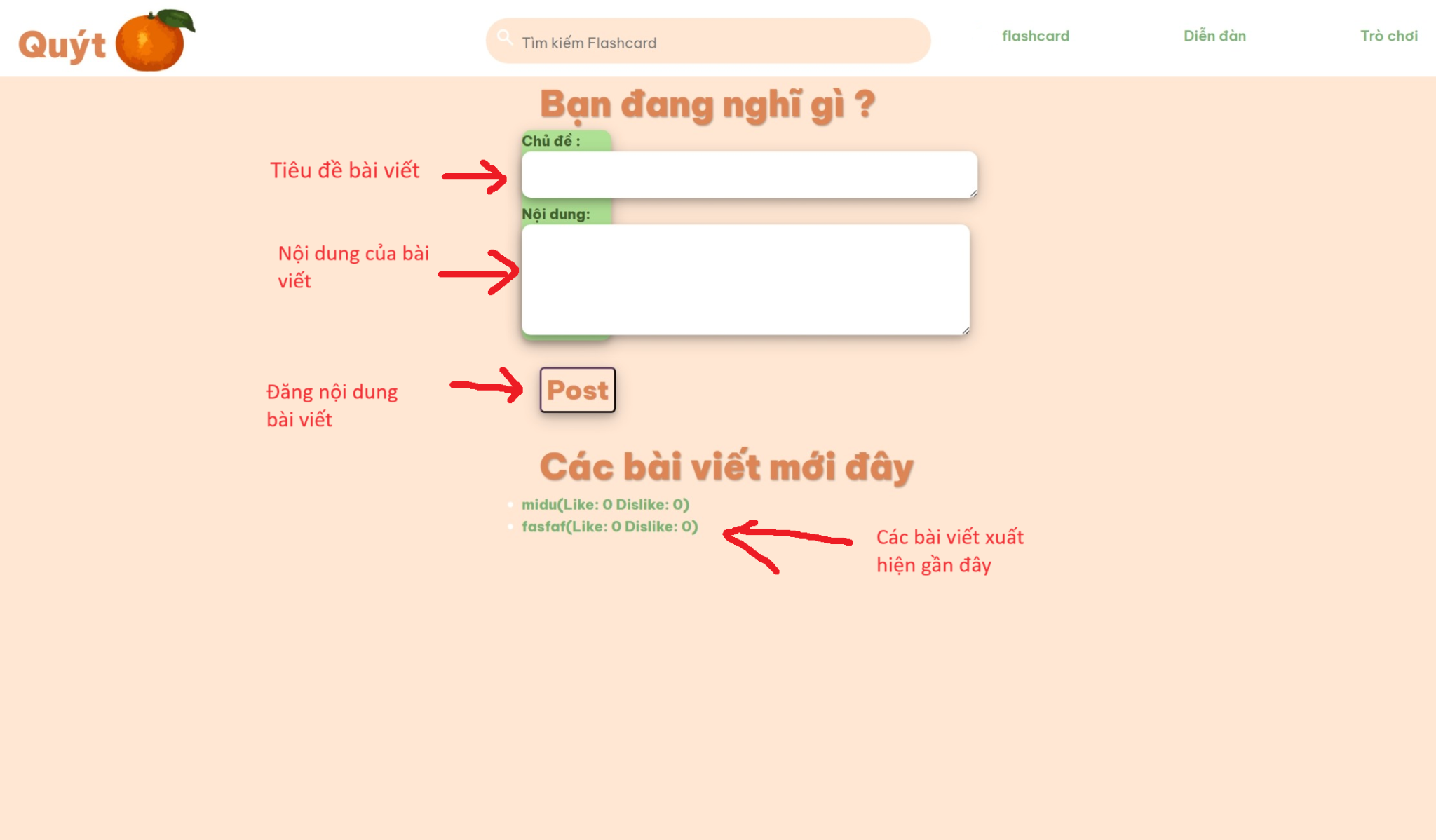
Để sử dụng Forum, người dùng sẽ click vào phần “Diễn đàn” trên thanh điều hướng. Sau đó, Web sẽ đưa người dùng đến giao diện của diễn đàn



15 - Giao diện nút diễn đàn

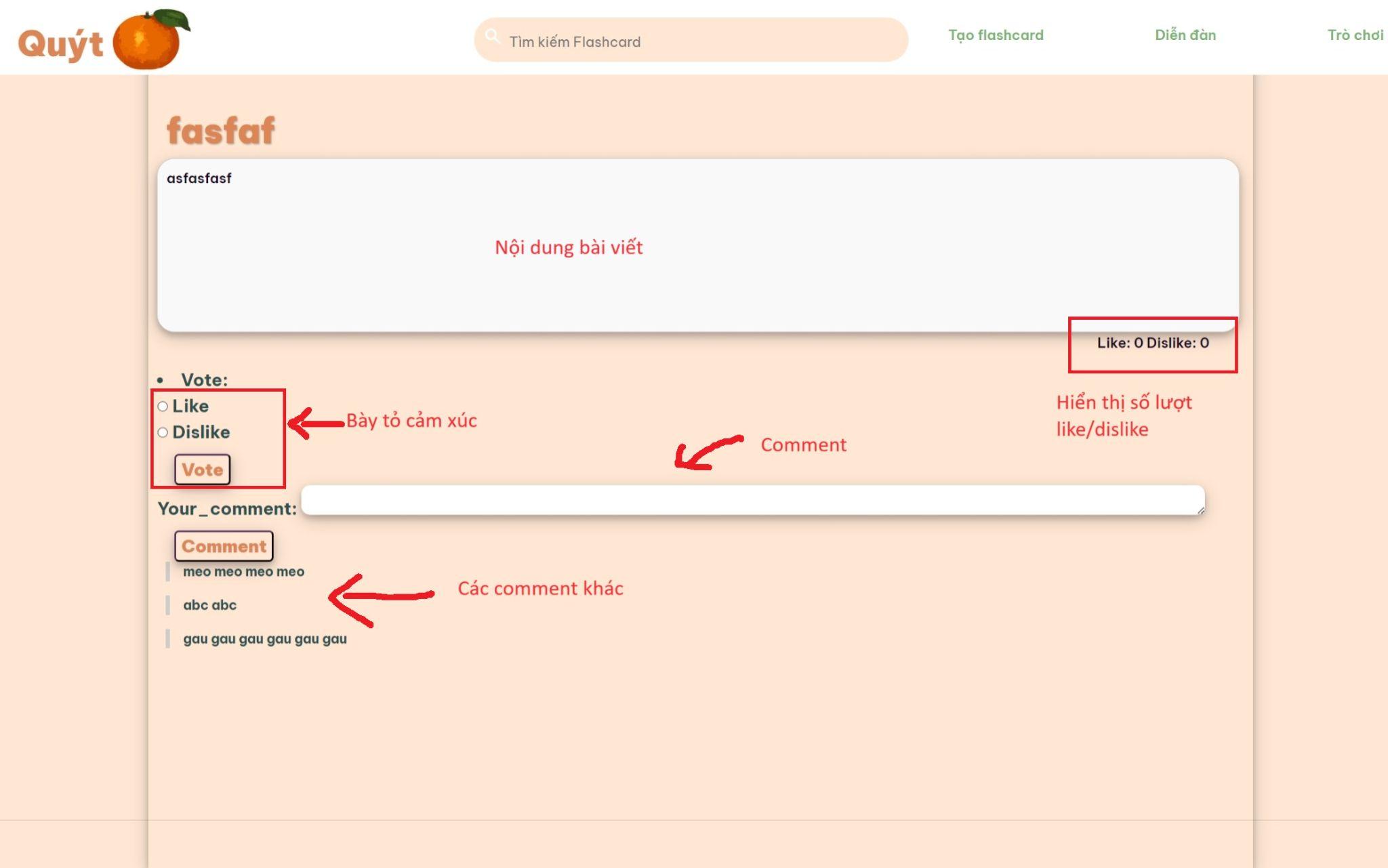
Sau khi click vào “Diễn đàn” trên thanh điều hướng, giao diện mới sẽ hiện ra, cho phép người dùng tạo thêm bài viết mới hoặc xem những bài viết của người khác.

* Trong trường hợp muốn tạo thêm bài viết mới, người dùng điền vào “Tiêu đề” và “Nội dung”, sau đó nhấn “Post” để đăng



16 - Giao diện thêm bài viết

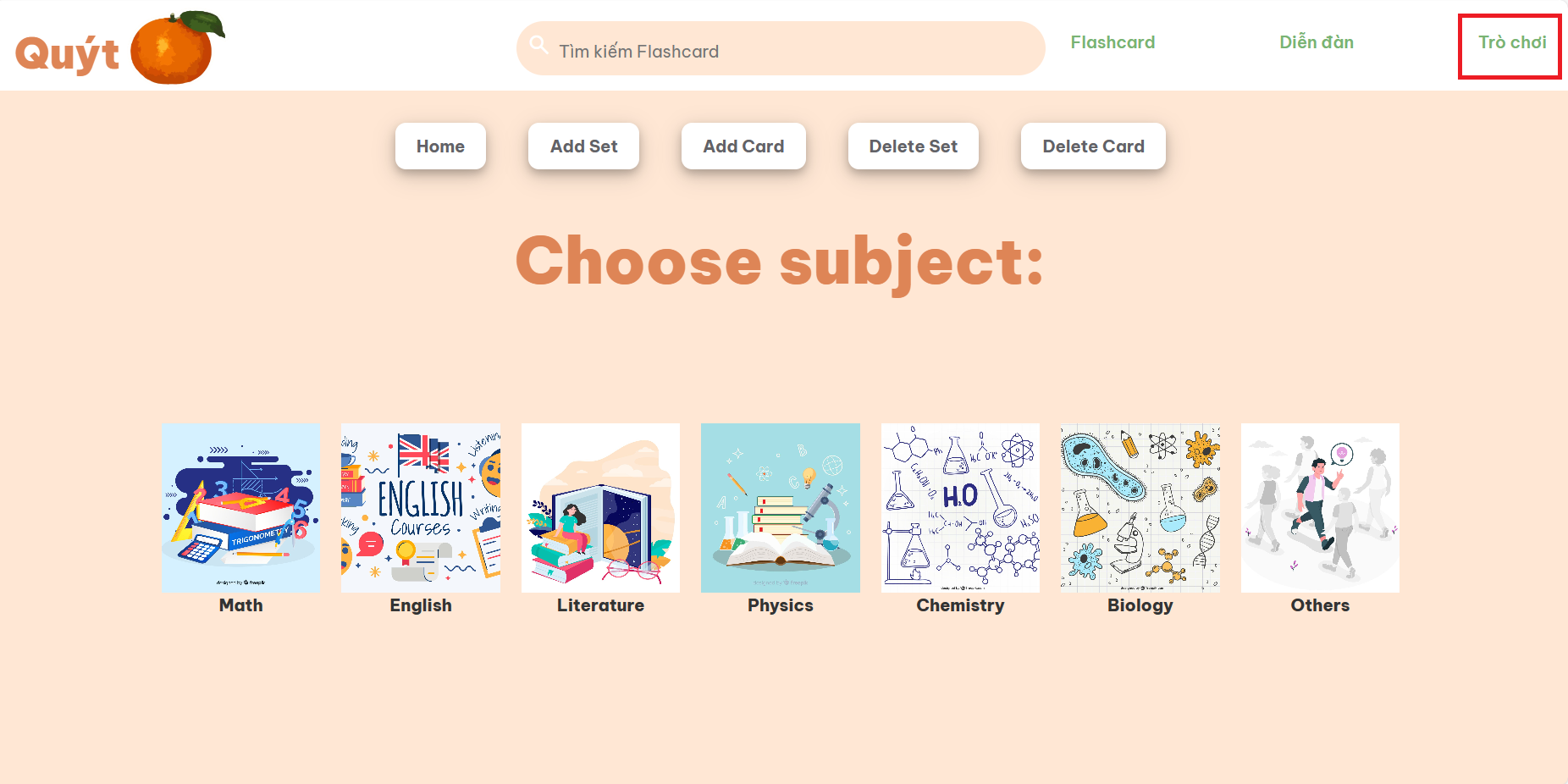
* Trong trường hợp muốn xem bài viết của người khác, click vào một bài bất kỳ trong phần “Các bài viết mới đây”, sau đó Web sẽ hiển thị bài viết mình muốn xem như hình bên dưới. Người dùng có thể tương tác trực tiếp với bài viết bằng cách comment/thả reaction



17 - Giao diện tương tác diễn đàn

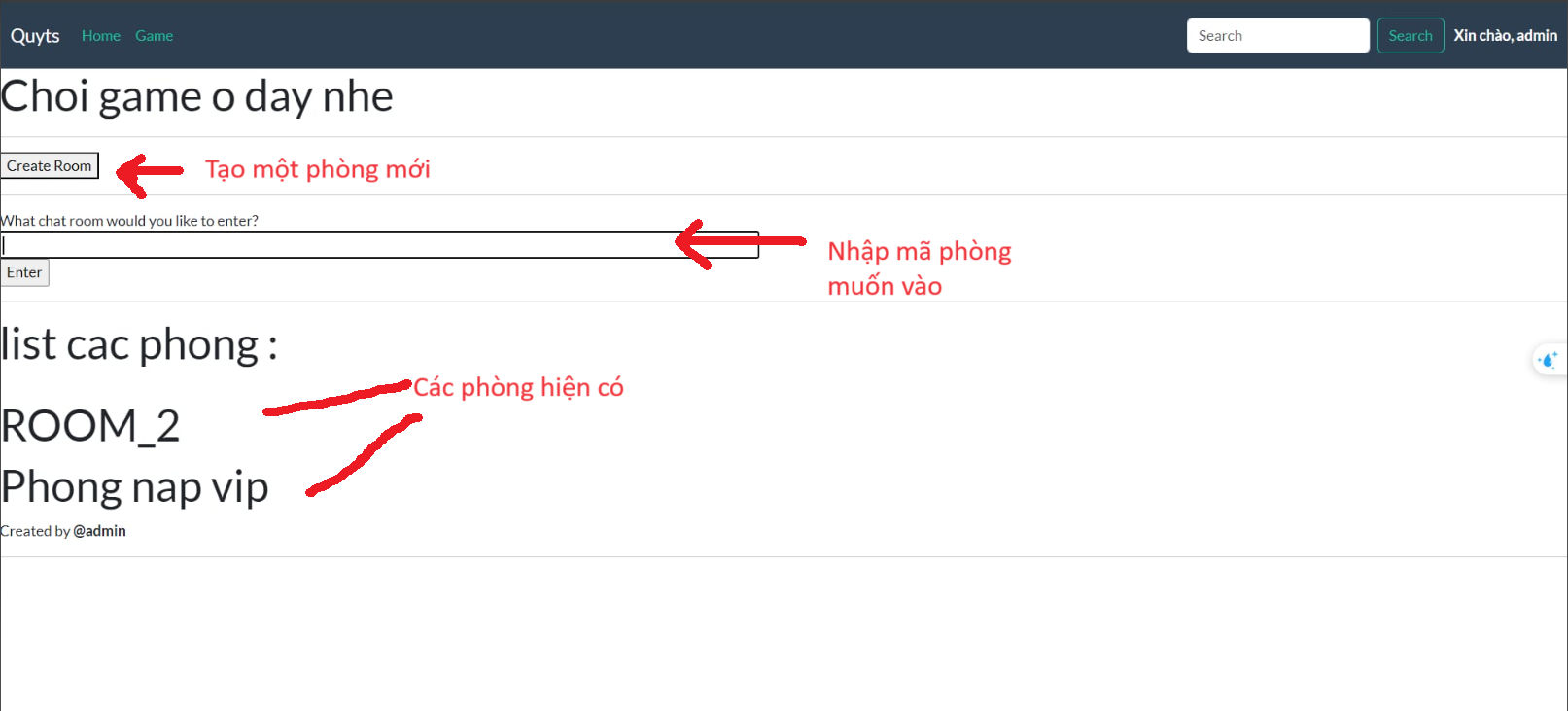
## Games:

Để có thể chơi game, người dùng có thể click vào phần “Trò chơi” ở góc phải trên cùng màn hình.



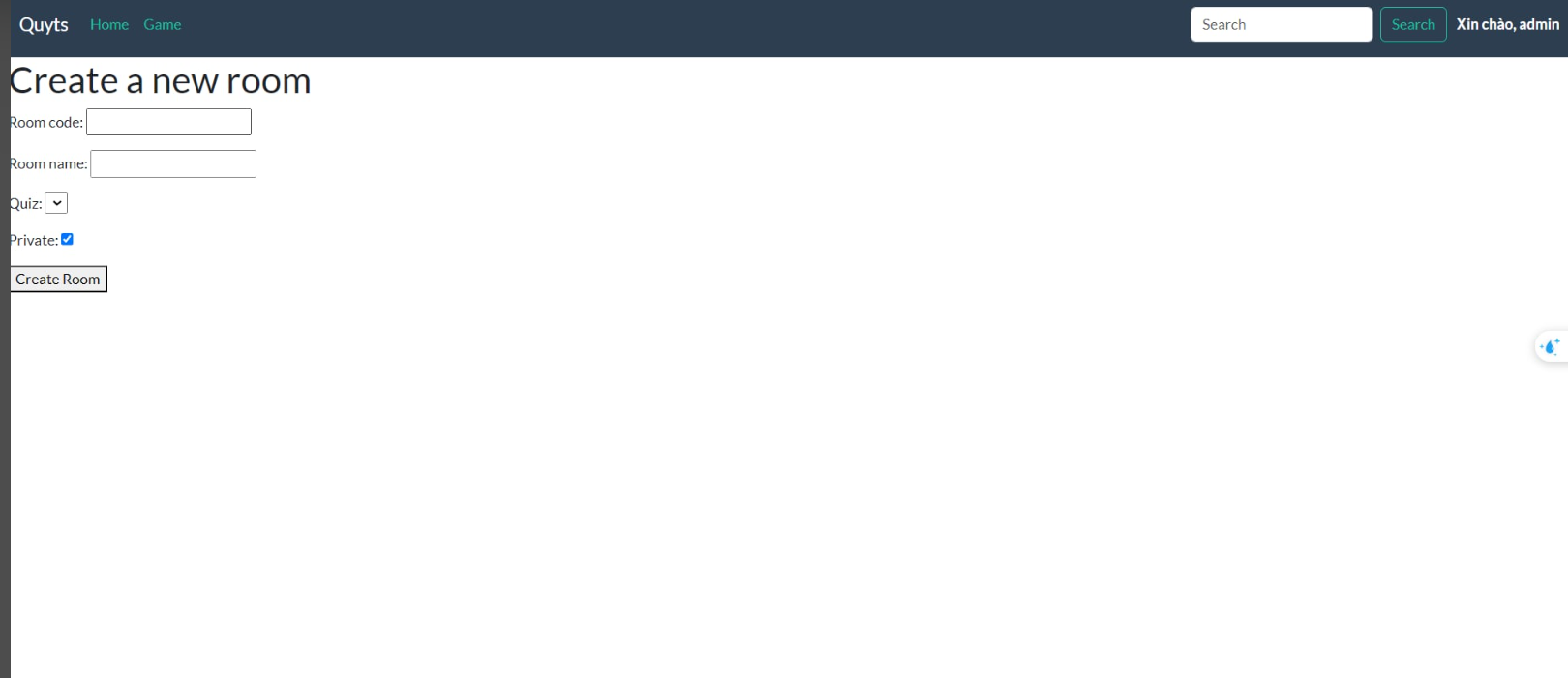
18 - Giao diện nút trò chơi

Sau khi click, một giao diện như bên dưới sẽ hiện ra. Người dùng có thể chọn “Create room” để tạo phòng mới hoặc nhập mã phòng muốn vào.



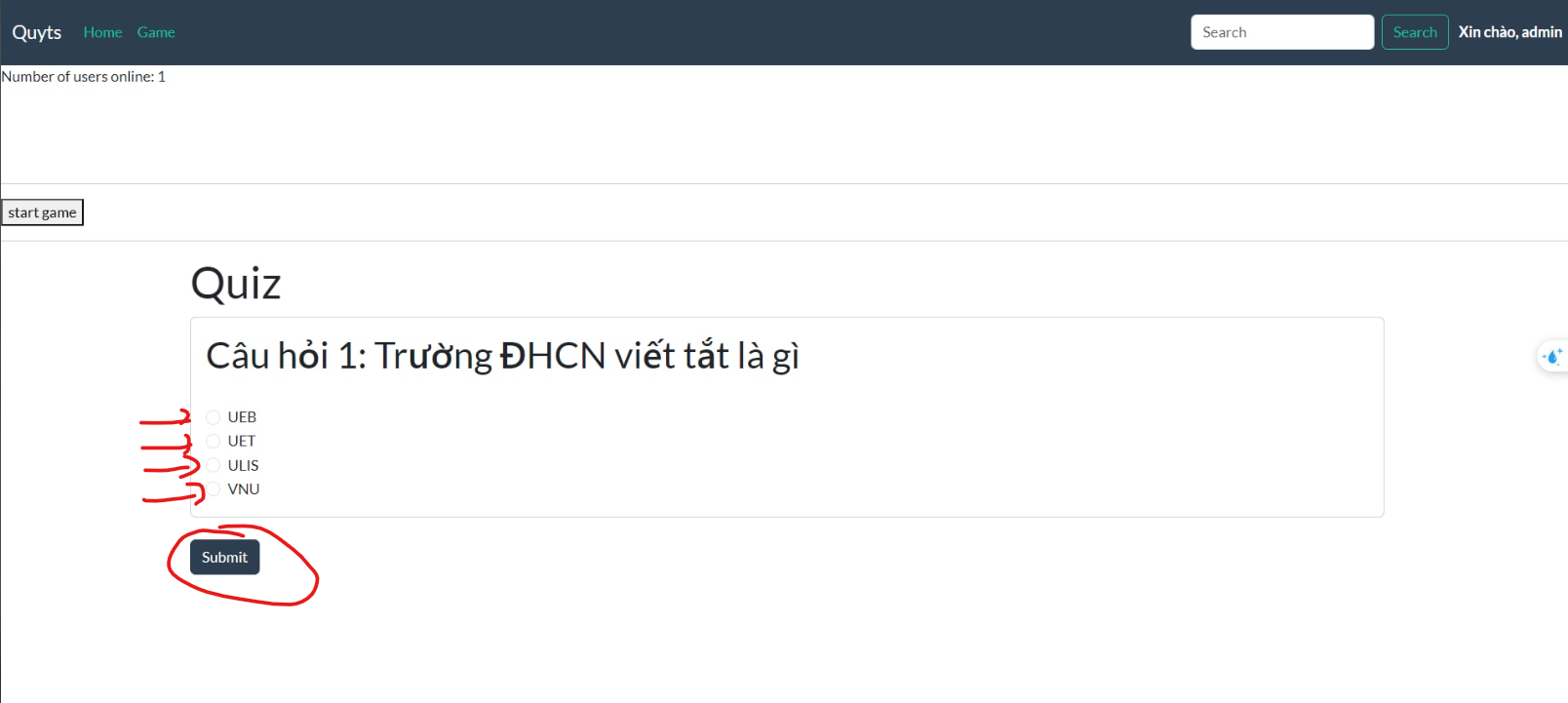
19 - Giao diện game

* **Tạo phòng mới**: Khi nhấn “Create Room”, một giao diện sẽ hiển thị như sau và người dùng sẽ nhập thông tin theo quy định của giao diện



20 - Giao diện tạo phòng

* **Nhập mã phòng và tham gia**: Khi nhập mã phòng và tham gia, người dùng sẽ thực hiện trả lời các câu hỏi của phòng chơi đó và nhấn “Submit” để nộp câu trả lời



21 - Giao diện gia nhập phòng

# Cách hoạt động của các chức năng chính

Dự án sử dụng mô hình theo hướng Client-Server do đó, việc giao tiếp giữa 2 bên sẽ thông qua việc gửi request và response. Phía Server sẽ quy định sẵn các URL (được hiểu như form request) để phía Client gửi yêu cầu lên.

## Đăng nhập / đăng ký:

* Người học sẽ đăng nhập qua chức năng đăng nhập của hệ thống. Hệ thống sẽ ghi nhận thông tin của người học rồi lưu vào trong cơ sở dữ liệu.
* Khi người dùng gửi yêu cầu đăng nhập thông qua một form HTML (trang đăng nhập), hệ thống sẽ nhận và xử lý yêu cầu này thông qua một hàm xử lý đặc biệt.
* Trong xử lý đăng nhập, hệ thống kiểm tra tính hợp lệ của thông tin đăng nhập (tên người học và mật khẩu).
* Sau khi đăng nhập thành công, người học sẽ được chuyển đến trang chủ của hệ thống.

## Flashcard:

Để sử dụng chức năng chính của Flashcard, người dùng có thể đăng nhập hoặc không (tùy theo mục đích sử dụng)

* **Đối với trường hợp không đăng nhập:**
  + Người dùng (Client) sẽ chỉ được phép thực hiện những request GET cho Server. Cụ thể là chỉ được xem nội dung của các bộ flashcard mà không thể thêm, sửa, xóa.
  + Phía Server sau khi nhận yêu cầu từ Client sẽ trả về bộ flashcard hoặc các flashcard tương ứng. Server sẽ tiến hành lấy tham số trong request và truy vấn trên cơ sở dữ liệu bao gồm một list các entities (việc truy vấn được xử lý trong phần flashcards/views.py). Sau khi truy vấn xong, Server sẽ tiến hành kết hợp kết quả trả về với form hiển thị HTML (trong phần flashcards/templates/flashcards/file.HTML) và hiển thị lên cho người dùng xem.
* **Đối với trường hợp đã đăng nhập:**
  + Người dùng (Client) sẽ được sử dụng toàn bộ chức năng liên quan đến flashcard (bao gồm xem, thêm, sửa, xóa).
  + Sau khi tiến hành đăng nhập để xác thực, phía Client sẽ gửi request lên server. Phía Server sau khi nhận được yêu cầu sẽ tự động xác định được đối tượng gửi request là ai.
  + Nếu trường hợp là request GET (xem và lấy thông tin các flashcard), quy trình xử lý sẽ giống như với trường hợp không đăng nhập.
  + Đối với trường hợp request POST (thêm, sửa, xóa), người dùng sẽ tiến hành nhập thông tin và gửi dữ liệu về cho phía server theo form HTML có sẵn. Server sau khi nhận request sẽ tự động xác nhận được request do ai gửi lên, từng miền thông tin là gì để tiến hành thực hiện yêu cầu.
    - Nếu request là thêm flashcard/bộ flashcard, phía Server sẽ tiến hành thực hiện câu lệnh CREATE entities tương ứng trong database. (việc xử lý database được thực hiện trong phần flashcards/views.py)
    - Nếu request là xóa flashcard/bộ flashcard, phía Server sẽ tiến hành thực hiện câu lệnh DELETE trong database. (việc xử lý database được thực hiện trong phần flashcards/views.py)

## Forum:

Tùy thuộc vào chức năng của forum mà người dùng có thể đăng nhập hoặc không.

* **Với chức năng xem bài viết:**

Người dùng hoàn toàn không cần đăng nhập mà vẫn có thể xem nội dung các bài viết bình thường.

* Người dùng sẽ chỉ thực hiện request GET cho Server - chỉ xem được nội dung, số lượt tương tác của bài viết và comment mà không thể tạo thêm bài viết, comment hay tương tác với chúng.
* Phía Server sau khi nhận request từ Client sẽ trả về bài viết tương ứng. Server sẽ lấy tham số trong request và truy vấn trên cơ sở dữ liệu - một danh sách các thực thể Post (bài viết). Sau khi truy vấn xong, Server sẽ tiến hành kết hợp kết quả trả về hiển thị HTML và hiển thị lên cho người dùng xem.
* **Với các chức năng tương tác / chỉnh sửa trong forum**:

Người dùng đăng nhập sẽ được sử dụng cả chức năng xem bài viết ở trên và sử dụng các chức năng thuộc phần này.

Sau khi xác thực người dùng hợp lệ, các chức năng sau mới được thực hiện:

* + **Đăng bài:** Người dùng sẽ thực hiện một request POST (thêm bài viết), để thực hiện request này, người dùng sẽ tiến hành nhập thông tin (title bài viết, nội dung bài viết) và gửi dữ liệu về phía Server theo form HTML có sẵn. Server sau khi nhận được request sẽ tự động xác nhận được request do ai gửi. Và phía Server sẽ tiến hành thực hiện câu lệnh CREATE thực thể Post trong database. Sau đó, trang sẽ được load lại và bài viết mới được up lên đầu trang.
  + **Comment bài viết:** Tương tự chức năng đăng bài, khi người dùng nhập thông tin vào form HTML có sẵn và gửi dữ liệu về phía Server thì comment của người dùng sẽ được đăng lên đầu danh sách comment. Cơ chế hoạt động tương tự như phần Đăng bài.

## Game:

Tùy thuộc vào chức năng của forum mà người dùng có thể đăng nhập hoặc không.

* **Với chức năng xem thông tin các phòng ở trạng thái công khai :** Người dùng hoàn toàn không cần đăng nhập mà vẫn có thể xem thông tin phòng công khai bình thường.
  + Người dùng sẽ chỉ thực hiện request GET cho Server - chỉ xem được tên phòng, mã phòng, chủ phòng và người dùng không thể tạo phòng hoặc tham gia phòng.
  + Phía Server sau khi nhận request từ Client sẽ trả các phòng public tương ứng. Server sẽ lấy tham số trong request và truy vấn trên cơ sở dữ liệu - một danh sách các thực thể Room (phòng). Sau khi truy vấn xong, Server sẽ tiến hành kết hợp kết quả trả về hiển thị HTML và hiển thị lên cho người dùng xem. (việc truy vấn được xử lý trong phần game/views.py)
* **Với các chức năng tạo phòng và tham gia phòng trong game**: Người dùng phải đăng nhập để sử dụng chức năng này. Sau khi xác thực người dùng hợp lệ, các chức năng sau mới được thực hiện:
  + **Tạo phòng** : Người dùng sẽ thực hiện một request POST (thêm phòng), để thực hiện request này, người dùng sẽ tiến hành nhập thông tin (tên phòng, chọn bộ câu hỏi,thuộc tính private hay public) và gửi dữ liệu về phía Server theo form HTML có sẵn. Server sau khi nhận được request sẽ tự động xác nhận được request do ai gửi. Và phía Server sẽ tiến hành thực hiện câu lệnh CREATE thực thể Room trong database. Sau đó, người dùng sẽ được định tuyến đến Room vừa được tạo.
  + **Tham gia phòng** : Người dùng sẽ thực hiện phương thức GET đến server, Phía Server sau khi nhận request từ Client sẽ dẫn người dùng đến phòng tương ứng. Server sẽ lấy tham số trong request và truy vấn trên cơ sở dữ liệu - một danh sách các thực thể Room (phòng). Sau khi truy vấn xong, Server sẽ tiến hành kết hợp kết quả trả về hiển thị HTML và hiển thị lên cho người dùng xem. (việc truy vấn được xử lý trong phần game/views.py)

# Thách thức và đề xuất giải pháp

## Thách thức / Giải pháp về thời gian phát triển

* **Thách thức**: Tình trạng thiếu thời gian thường bắt nguồn từ việc đánh giá không chính xác về độ phức tạp của các nhiệm vụ, dẫn đến sự trễ hẹn trong việc hoàn thành dự án và gây ra hiệu ứng domino làm ảnh hưởng đến quá trình phát triển về sau.
* **Giải pháp**: Liên tục đánh giá tiến độ dự án theo chu kỳ (khoảng 1-2 tuần/lần) để sớm giải quyết, khắc phục những sai lệch càng sớm càng tốt

## Thách thức / Giải pháp về thực hiện đúng yêu cầu dự án

* **Thách thức**: Việc hiểu sai yêu cầu dự án là điều hoàn toàn có thể xảy ra. Từ đó dẫn đến sai sót trong quá trình phát triển, gây lãng phí thời gian và chi phí
* **Giải pháp**: Liên tục đưa ra những bản demo, cuộc thảo luận giữa các thành viên để xác định đúng hướng phát triển

## Thách thức / Giải pháp về bảo mật

* **Thách thức**: Sản phẩm vẫn còn thô sơ và chưa có các biện pháp bảo mật, có thể gây ra các vấn đề nghiêm trọng như mất dữ liệu, vi phạm quyền riêng tư.
* **Giải pháp**: Một số phương pháp có thể được áp dụng bao gồm sử dụng các phần mềm quản lý dữ liệu từ các bên thứ ba uy tín và tuân thủ các nguyên tắc cốt lõi về an toàn thông tin.

## Thách thức / Giải pháp liên quan đến nội dung bẩn

* **Thách thức**: Sản phẩm vẫn còn thô sơ, chưa có các biện pháp kiểm duyệt nội dung và từ ngữ
* **Giải pháp**: Sau này có thể xây dựng hệ thống lọc từ ngữ nhạy cảm và các hình phạt phạt đối với những trường hợp vi phạm