

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



Đồ án tổng hợp - hướng hệ thống thông tin - CO3015

Báo cáo

**XÂY DỰNG WEBSITE TRAINING NHÂN SỰ CHO
DOANH NGHIỆP**

Giảng viên hướng dẫn: ThS. Trần Thị Quế Nguyệt

Sinh viên thực hiện: 2014486 - Đậu Xuân Thành

TP. Hồ Chí Minh, 08/2023



Mục lục

1. Giới thiệu đề tài	4
1.1. Phân tích yêu cầu	4
1.1.1. Yêu cầu chức năng	4
2. Công nghệ sử dụng	5
2.1. Phía client (Front - end)	5
2.2. Phía server (Back - end)	6
2.3. Khác	8
3. Database Design	9
3.1. Thiết kế mô hình thực thể mối liên kết (ERD Model)	9
3.2. Lược đồ cơ sở dữ liệu quan hệ (Database Schema)	11
3.3. Cơ sở dữ liệu trong PostgreSQL	11
4. Hiện thực	14
4.1. Tổng quan kiến trúc hệ thống	14
4.2. Giao diện	15
4.2.1. Đăng nhập, đăng ký	15
4.2.2. Trang chủ	16
4.2.3. Bài đăng	17
4.2.4. Admin Panel	18
Tài liệu tham khảo	23

Danh mục hình ảnh

Hình 1: ReactJS là một thư viện JavaScript phổ biến hiện nay.	5
Hình 2: MUI là một thư viện của ReactJS cung cấp giao diện hiện đại.	6
Hình 3: PostgreSQL là một hệ quản trị cơ sở dữ liệu ổn định, mạnh mẽ và linh hoạt.	7
Hình 4: Flask là một thư viện Python giúp phát triển web một cách đơn giản, nhanh chóng	7
Hình 5: Thực thể người dùng (users)	9
Hình 6: Thực thể Chủ đề (topics)	9
Hình 7: Thực thể Bài đăng (posts)	9
Hình 8: Thực thể Câu hỏi (questions)	9
Hình 9: Thực thể Bài kiểm tra (tests)	10
Hình 10: Mối liên kết giữa Topics và Posts	10
Hình 11: Lược đồ thực thể mối liên kết cho website	10
Hình 12: Lược đồ cơ sở dữ liệu quan hệ của website	11
Hình 13: Topics Schema	12
Hình 14: Users Schema	13
Hình 15: Tổng quan kiến trúc hệ thống	14
Hình 16: Giao diện khi đăng nhập	15
Hình 17: Giao diện khi đăng ký	15
Hình 18: Trang chủ	16
Hình 19: Tìm kiếm chủ đề	17
Hình 20: Mỗi bài đăng được Admin tạo bằng html editor	17
Hình 21: Dialog hiển thị các câu hỏi dưới dạng trắc nghiệm	18
Hình 22: Quản lý người dùng	18
Hình 23: Xem thông tin các chủ đề hiện có	19
Hình 24: Tạo chủ đề mới	19
Hình 25: Xem thông tin bài đăng của một chủ đề	20



Hình 26: Tạo bài đăng cho một chủ đề	20
Hình 27: Chỉnh sửa bài đăng	21
Hình 28: Xem thông tin câu hỏi	21
Hình 29: Tạo câu hỏi mới	22



1. Giới thiệu đề tài

Trong hoạt động của mọi tổ chức, doanh nghiệp đều đặt ra các quy định, luật lệ để có doanh nghiệp có thể vận hành một cách có tổ chức, chuyên nghiệp. Ngoài ra, trong từng lĩnh vực, chuyên môn khác nhau đều có nhu cầu huấn luyện nhân sự của họ để nâng cao hiệu suất làm việc, kinh doanh. Vì vậy, đòi hỏi phải xây dựng một hệ thống để giúp cho các tổ chức này phổ biến quy định một cách nhanh chóng hơn hoặc giúp kiểm tra kiến thức của nhân viên theo giai đoạn. Ở đây, chúng ta sẽ xây dựng một website dùng để huấn luyện (training) nhân sự của một doanh nghiệp.

1.1. Phân tích yêu cầu

1.1.1. Yêu cầu chức năng

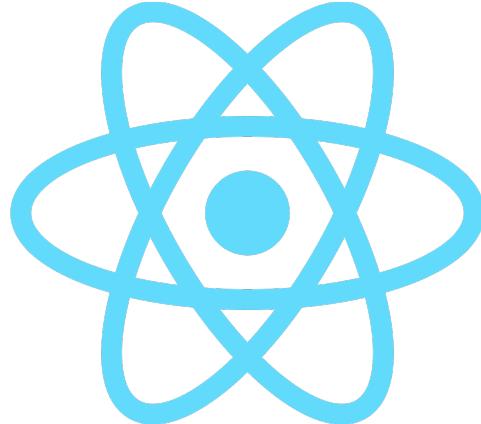
- Admin có thể tạo bài đăng (post) theo các chủ đề (topic), mỗi chủ đề gồm nhiều bài đăng (post) có thứ tự hiển thị.
- Mỗi bài đăng có các câu hỏi(question) liên quan để người học trả lời câu hỏi dạng multiple choice question.
- Mỗi chủ đề có các thông tin: ảnh đại diện, tên chủ đề, từ khoá chủ đề, độ khó, và thông tin dẫn xuất số bài đăng của chủ đề đó.
- Mỗi bài đăng có các nội dung như tiêu đề lớn, mô tả ngắn, nội dung bài đăng (nhập html editor), ảnh banner bài đăng, thời gian tạo, thời gian chỉnh sửa, từ khoá liên quan, link short_url (slug), số phút đọc ước tính, số câu hỏi sinh ngẫu nhiên (từ danh sách câu hỏi liên quan bài đăng đã tạo).
- Người dùng là nhân viên công ty, có thể đăng ký tài khoản, nhưng được sử dụng trang web chỉ khi được admin approve tài khoản. Có các chức năng khác như quên mật khẩu, đổi mật khẩu (gửi qua email xác nhận)
- User khi vào trang web sẽ thấy danh sách các chủ đề, khi vào 1 chủ đề sẽ xem được danh mục các bài đăng với thứ tự hiển thị như admin đã thiết lập. Khi xem 1 bài đăng, user có thể có các chức năng Next (đi tới bài đăng tiếp) Previous (bài đăng trước). Nếu là bài đăng đầu tiên hay cuối cùng thì các nút tương ứng sẽ bị mờ.
- Khi di chuyển qua bài đăng kế tiếp thì user phải trả lời đúng hết các câu hỏi của bài đăng hiện tại. Nếu các câu hỏi của bài đăng hiện tại đã trả lời cho lần xem trước đó thì không cần phải trả lời lại. Khi hiển thị câu hỏi, hệ thống kiểm tra số câu hỏi cho bài đăng đó là bao nhiêu, và lấy ra ngẫu nhiên số câu hỏi từ danh sách câu hỏi của bài đăng để cho người dùng trả lời.
- Hệ thống ghi nhận và hiển thị bài đăng nào user đã hoàn thành (nghĩa là đã trả lời hết câu hỏi), chưa hoàn thành (nghĩa là câu hỏi trả lời chưa đúng hoặc chưa trả lời câu hỏi nào). Người dùng có thể xem lại câu hỏi và kết quả trả lời của mình ở mỗi bài đăng.
- Khi xem danh sách các chủ đề ở trang chủ, user sẽ thấy các thông tin ảnh đại diện của chủ đề, độ khó chủ đề, tổng số lượng bài đăng trong chủ đề, số bài đăng đã hoàn thành. Có thể tìm kiếm chủ đề dựa vào tên chủ đề, lọc theo độ khó, và chưa hoàn thành/ hoàn thành.

2. Công nghệ sử dụng

2.1. Phía client (Front - end)

ReactJS

ReactJS là một thư viện JavaScript phổ biến được sử dụng để xây dựng giao diện người dùng động và tương tác trên các ứng dụng web. Với ReactJS, chúng ta có thể tạo ra các thành phần UI đơn giản hoặc phức tạp, và sau đó kết hợp chúng lại để tạo thành các ứng dụng phong phú.



Hình 1: ReactJS là một thư viện JavaScript phổ biến hiện nay.

Một số ưu điểm của ReactJS bao gồm:

- **Component-based:** ReactJS sử dụng cấu trúc dựa trên thành phần, cho phép chúng ta tái sử dụng các thành phần UI và xây dựng giao diện phức tạp bằng cách kết hợp các thành phần nhỏ hơn. Điều này giúp cải thiện tính module, rõ ràng và dễ bảo trì của mã nguồn.
- **Virtual DOM:** ReactJS sử dụng một DOM ảo để quản lý và cập nhật giao diện người dùng. Khi có sự thay đổi trong dữ liệu, ReactJS so sánh DOM ảo và DOM thực tế, và chỉ cập nhật những phần thay đổi thực sự. Điều này giúp tăng hiệu suất ứng dụng và giảm tải cho trình duyệt.
- **High Performance:** Nhờ cơ chế cập nhật thông minh và sử dụng DOM ảo, ReactJS có khả năng xử lý hiệu quả các ứng dụng có giao diện phức tạp và dữ liệu động. Ngoài ra, ReactJS cũng cung cấp các công cụ tối ưu hóa như React.memo, PureComponent để giảm thiểu việc cập nhật không cần thiết và tăng tốc độ render.
- **Hỗ trợ cho cả client-side và server-side:** ReactJS cho phép chúng ta xây dựng ứng dụng đơn trang (single-page applications) trên client-side, nhưng cũng hỗ trợ server-side rendering (SSR). SSR giúp tăng tốc độ tải trang ban đầu và cải thiện trải nghiệm người dùng.
- **Hệ sinh thái mạnh mẽ:** ReactJS có một hệ sinh thái phát triển đa dạng và phong phú, với nhiều thư viện và công cụ hỗ trợ như React Router, Redux, MobX, Styled Components, và nhiều hơn nữa. Điều này giúp chúng ta xây dựng ứng dụng theo cách tốt nhất và lựa chọn các công nghệ phù hợp với dự án của mình.

Sau khi tìm hiểu ưu, nhược điểm của ReactJS, sinh viên quyết định sử dụng thư viện này làm công nghệ chính để phát triển trang web ở phía client (front - end).

Material UI (MUI) [1]

Để thiết kế giao diện sử dụng ReactJS, sinh viên quyết định sử dụng thư viện Marterial-UI của ReactJS, để nâng cao trải nghiệm người dùng, tối ưu hóa những ưu điểm của ReactJS.

Material-UI là một thư viện giao diện người dùng cho ReactJS, được xây dựng dựa trên nguyên tắc thiết kế của Google Material Design. Nó cung cấp các thành phần UI và các phong cách mô phỏng

hầu hết các yếu tố giao diện người dùng của Material Design, giúp chúng ta xây dựng ứng dụng web với giao diện hấp dẫn và chuyên nghiệp.



Hình 2: MUI là một thư viện của ReactJS cung cấp giao diện hiện đại.

Một số ưu điểm của Material-UI bao gồm:

- **Thiết kế theo nguyên tắc Material Design:** Với Material-UI, chúng ta có thể sử dụng các thành phần UI đã được thiết kế sẵn theo nguyên tắc của Google Material Design, bao gồm nút, thanh điều hướng, biểu mẫu, bảng dữ liệu và nhiều hơn nữa. Điều này giúp tạo ra giao diện người dùng đẹp mắt, mượt mà và dễ sử dụng.
- **Tùy chỉnh và tái sử dụng dễ dàng:** Material-UI cho phép chúng ta tùy chỉnh các thành phần UI để phù hợp với nhu cầu thiết kế của chúng ta. Chúng ta có thể thay đổi màu sắc, kích thước, kiểu dáng và các thuộc tính khác của các thành phần. Ngoài ra, Material-UI cung cấp các thành phần tái sử dụng, giúp chúng ta tiết kiệm thời gian và công sức trong việc xây dựng giao diện người dùng.
- **Hỗ trợ tương thích đa trình duyệt:** Material-UI được thiết kế để tương thích với nhiều trình duyệt khác nhau, bao gồm cả trình duyệt di động. Điều này giúp đảm bảo rằng ứng dụng của chúng ta sẽ hoạt động một cách nhất quán trên các nền tảng khác nhau mà không cần quá nhiều công việc tùy chỉnh.
- **Hệ sinh thái mạnh mẽ:** Material-UI có một hệ sinh thái đa dạng và phong phú, bao gồm nhiều thành phần, mẫu thiết kế, tiện ích và các công cụ hỗ trợ. Chúng ta có thể tìm thấy sẵn một loạt các thành phần và giải pháp để giúp chúng ta xây dựng giao diện người dùng chất lượng cao một cách nhanh chóng và dễ dàng.

2.2. Phía server (Back - end)

PostgreSQL [2]

PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở và mạnh mẽ. Nó được phát triển với sự chú trọng vào tính đồng nhất, bảo mật và khả năng mở rộng. PostgreSQL hỗ trợ các tính năng SQL tiên tiến, cung cấp một môi trường đáng tin cậy để lưu trữ và quản lý dữ liệu.



Hình 3: PostgreSQL là một hệ quản trị cơ sở dữ liệu ổn định, mạnh mẽ và linh hoạt.

Một số ưu điểm của PostgreSQL bao gồm:

- **Độ tin cậy cao:** PostgreSQL được xây dựng với sự chú trọng đến tính đồng nhất và bảo mật. Nó sử dụng các cơ chế ghi nhật ký (write-ahead logging) để đảm bảo dữ liệu được bảo vệ và không mất mát trong trường hợp xảy ra sự cố. PostgreSQL cũng hỗ trợ các công nghệ như sao lưu, phục hồi và replica để đảm bảo tính sẵn sàng cao và khả năng khôi phục sau sự cố.
- **Hỗ trợ đa nền tảng:** PostgreSQL có thể chạy trên nhiều nền tảng, bao gồm Linux, Windows và macOS. Điều này cho phép bạn triển khai và vận hành PostgreSQL trên các hệ thống khác nhau một cách linh hoạt.
- **Tính mở rộng:** PostgreSQL có khả năng mở rộng tốt, cho phép bạn mở rộng cấu trúc cơ sở dữ liệu theo nhu cầu của ứng dụng. Bạn có thể tăng cường hiệu suất và khả năng chịu tải của PostgreSQL bằng cách sử dụng các công nghệ như phân vùng, chỉ mục và replica.
- **Tính linh hoạt và mạnh mẽ:** PostgreSQL cung cấp một loạt các tính năng tiên tiến và mạnh mẽ, bao gồm hỗ trợ truy vấn phức tạp, khóa dữ liệu đa mức, xử lý giao dịch an toàn và hỗ trợ các kiểu dữ liệu đa dạng. Nó cũng hỗ trợ các ngôn ngữ lập trình phổ biến như Python, Java, C/C++, và nhiều ngôn ngữ khác.
- **Cộng đồng phát triển đông đảo:** PostgreSQL có một cộng đồng phát triển rộng lớn, với nhiều nguồn tài nguyên, tài liệu và hỗ trợ từ cộng đồng. Việc có một cộng đồng phát triển mạnh mẽ giúp đảm bảo rằng PostgreSQL được cải thiện và bảo trì một cách liên tục.

PostgreSQL là một hệ quản trị cơ sở dữ liệu ổn định, mạnh mẽ và linh hoạt. Với tính tin cậy cao, tính mở rộng, và nhiều tính năng tiên tiến, PostgreSQL là một lựa chọn phổ biến cho việc lưu trữ và quản lý dữ liệu trong các ứng dụng phức tạp. Vì vậy, ở sinh viên sử dụng PostgreSQL để lưu trữ và quản lý cơ sở dữ liệu của trang web này.

Flask [3]

Ở trong dự án này, chúng ta sử dụng Flask để xử lý logic ở phía server.

Flask là một framework micro web phát triển bằng Python, nhẹ nhàng và dễ sử dụng. Với Flask, bạn có thể nhanh chóng xây dựng các ứng dụng web nhỏ đến trung bình mà không cần một số lượng lớn các thành phần và tính năng phức tạp. Flask tập trung vào sự đơn giản, linh hoạt và dễ mở rộng, làm cho việc phát triển ứng dụng web trở nên dễ dàng và nhanh chóng.



Hình 4: Flask là một thư viện Python giúp phát triển web một cách đơn giản, nhanh chóng



Một số ưu điểm của Flask bao gồm:

- **Đơn giản và dễ sử dụng:** Flask có cấu trúc đơn giản và ít đặt quy tắc, giúp người dùng nắm bắt nhanh chóng và phát triển ứng dụng một cách dễ dàng. Nó không có các yêu cầu phức tạp và cho phép bạn tùy chỉnh theo nhu cầu cụ thể của dự án.
- **Framework micro:** Flask được thiết kế như một framework micro, điều này có nghĩa là nó chỉ cung cấp các thành phần cơ bản cần thiết để xây dựng ứng dụng web. Tuy nhiên, Flask cũng cho phép bạn mở rộng chức năng bằng cách sử dụng các tiện ích và các extension của cộng đồng.
- **Linh hoạt và tùy chỉnh:** Flask cho phép bạn tùy chỉnh và linh hoạt trong việc xây dựng ứng dụng web. Bạn có thể lựa chọn các thành phần cần thiết cho dự án của mình và sử dụng các extension để bổ sung các tính năng như xác thực, cơ sở dữ liệu, giao diện người dùng, và nhiều hơn nữa.
- **Hỗ trợ tốt cho Python:** Flask được xây dựng bằng Python và tương thích tốt với các thư viện và công cụ của Python. Điều này cho phép bạn sử dụng các thư viện Python phổ biến để xử lý các tác vụ phức tạp và tích hợp dễ dàng với hệ thống tồn tại.
- **Cộng đồng phát triển đông đảo:** Flask có một cộng đồng phát triển lớn, với nhiều nguồn tài nguyên, ví dụ như tài liệu, các ví dụ mã nguồn và các extension được xây dựng bởi cộng đồng. Điều này giúp bạn tìm kiếm giải pháp, hỗ trợ và nâng cao kỹ năng phát triển của mình.

Flask là một lựa chọn tuyệt vời cho việc phát triển ứng dụng web đơn giản và linh hoạt. Với tính đơn giản, dễ sử dụng và khả năng tùy chỉnh, Flask giúp bạn xây dựng ứng dụng web nhanh chóng và dễ dàng.

2.3. Khác

Ngoài những công nghệ chính trên, sinh viên còn sử dụng các công cụ hỗ trợ khác trong quá trình phát triển trang web để nâng cao hiệu suất như:

- Postman: Test API phía back-end.
- Google DevTools: Test phía front-end.
- Toggle Pesticide: Kiểm tra cấu trúc của giao diện người dùng.

...

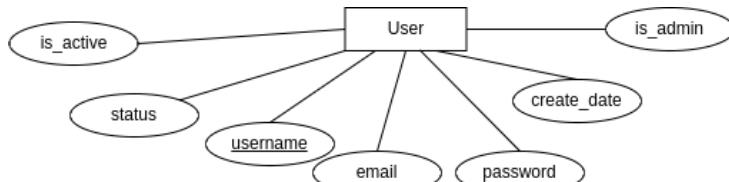
3. Database Design

Từ các yêu cầu về dữ liệu, chúng ta cần phân tích để thiết kế cơ sở dữ liệu.

3.1. Thiết kế mô hình thực thể mối liên kết (ERD Model)

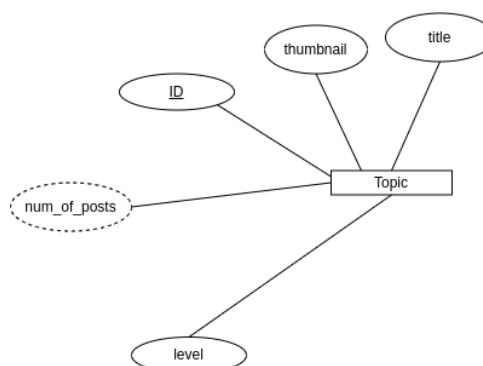
Từ các yêu cầu về dữ liệu, có thể nhận thấy có 5 thực thể chính trong hệ thống dữ liệu của ứng dụng:

- Người dùng - **Users** (bao gồm: Admin và Nhân viên)



Hình 5: Thực thể người dùng (users)

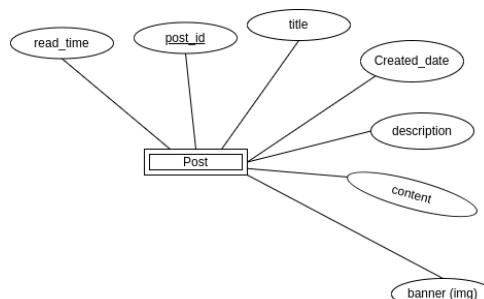
- Chủ đề - **Topics**



Hình 6: Thực thể Chủ đề (topics)

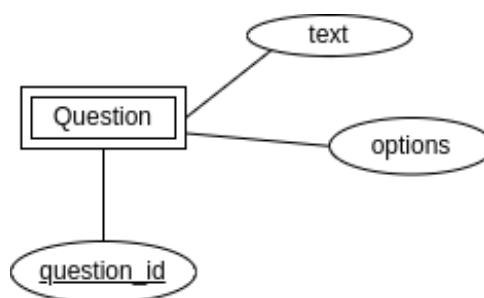
- Bài đăng - **Posts**

Đây là một thực thể yếu vì nó phụ thuộc vào thực thể **Chủ đề**



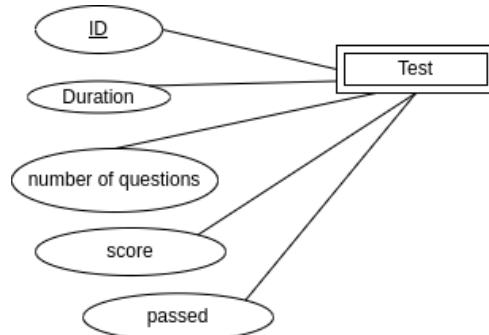
Hình 7: Thực thể Bài đăng (posts)

- Câu hỏi - **Questions**



Hình 8: Thực thể Câu hỏi (questions)

- Bài kiểm tra - **Tests** (được tạo khi người dùng trả lời các câu hỏi)

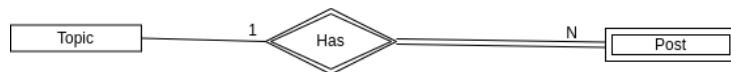


Hình 9: Thực thể Bài kiểm tra (tests)

Phân tích các mối quan hệ trong hệ thống:

- Các Chủ đề bao gồm các bài đăng:

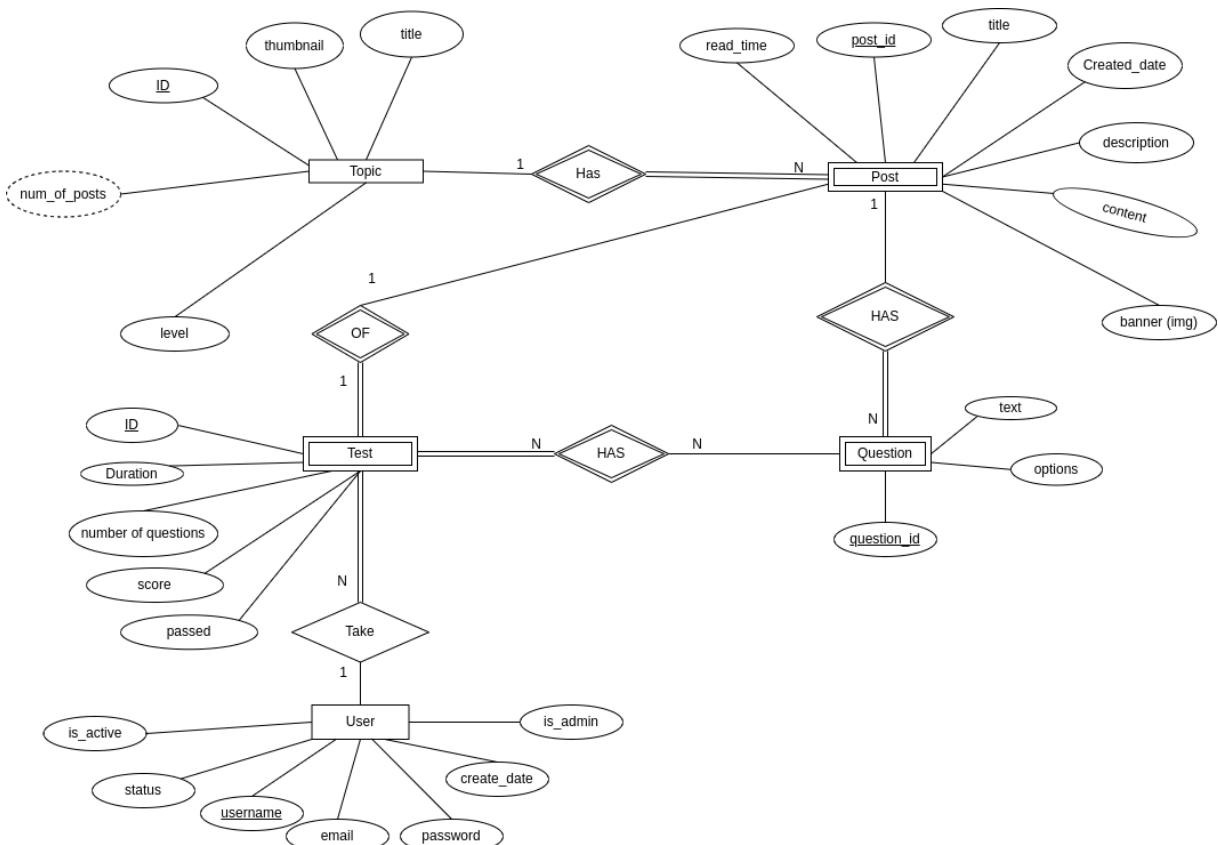
Mỗi **Chủ đề** bao gồm nhiều **Bài đăng**, nhưng mỗi **Bài đăng** chỉ thuộc về một **Chủ đề** cụ thể.



Hình 10: Mối liên kết giữa Topics và Posts

...

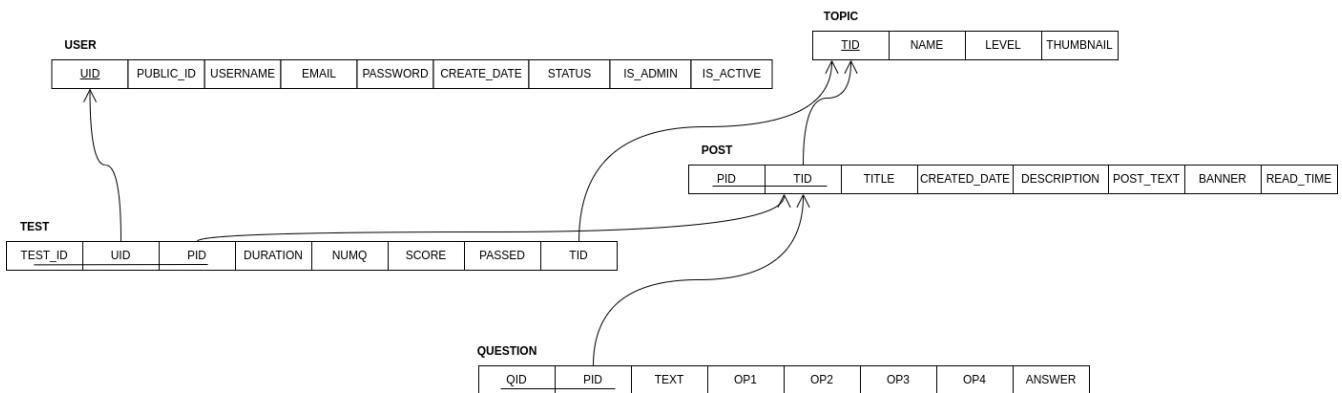
Tổng quan lược đồ thực thể mối liên kết:



Hình 11: Lược đồ thực thể mối liên kết cho website

3.2. Lược đồ cơ sở dữ liệu quan hệ (Database Schema)

Từ lược đồ thực thể mỗi liên kết, chúng ta có thể ánh xạ thành một lược đồ cơ sở dữ liệu quan hệ.



Hình 12: Lược đồ cơ sở dữ liệu quan hệ của website

3.3. Cơ sở dữ liệu trong PostgreSQL

Sau có lược đồ cơ sở dữ liệu quan hệ, chúng ta bắt tay vào thiết kế cơ sở dữ liệu trong PostgreSQL:

```
--- Bảng Topics
CREATE TABLE
    Topics (
        topic_id SERIAL,
        topic_name VARCHAR(255) NOT NULL,
        topic_level INT NOT NULL,
        thumbnail VARCHAR(255) NOT NULL
    );

--- Bảng Users
CREATE TABLE
    Users (
        uuid SERIAL NOT NULL,
        username VARCHAR(255) NOT NULL UNIQUE,
        email VARCHAR(100) NOT NULL,
        passwd VARCHAR(255),
        created_date DATE,
        is_active BOOLEAN,
        is_admin BOOLEAN,
        u_avatar VARCHAR(255)
    );

.....
--- Các ràng buộc
ALTER TABLE Topics ADD CONSTRAINT pk_topic PRIMARY KEY (topic_id);

ALTER TABLE Users ADD CONSTRAINT pk_user PRIMARY KEY (uuid);

.....
--- Các procedures, functions
.....
--- Hàm lấy số lượng bài đăng mà một user đã hoàn thành
CREATE OR REPLACE FUNCTION get_passed_post_count_by_user(user_id INT)
RETURNS TABLE (
    topic_id INT,
    topic_name VARCHAR(255),
    uuid INT,
    passed_post_count INT
```



```
)  
AS $$  
BEGIN  
RETURN QUERY  
SELECT t.topic_id, t.topic_name, pt.uuid, COUNT(pt.post_id)::INT AS passed_post_count  
FROM topics t  
JOIN posts p ON t.topic_id = p.topic_id  
JOIN tests pt ON p.post_id = pt.post_id  
WHERE pt.passed = true AND pt.uuid = user_id  
GROUP BY t.topic_id, t.topic_name, pt.uuid;  
END;  
$$ LANGUAGE plpgsql;
```

....

Các relation đang có trong cơ sở dữ liệu trainning-website:

```
trainning-website-# \dt  
List of relations  
Schema | Name | Type | Owner  
-----+-----+-----+-----  
public | posts | table | thanhdxn  
public | question | table | thanhdxn  
public | tests | table | thanhdxn  
public | topics | table | thanhdxn  
public | users | table | thanhdxn  
(5 rows)
```

Xem cấu trúc schema của các bảng trong cơ sở dữ liệu, ví dụ:

Bảng topics:

```
trainning-website-# \d topics
```

```
trainning-website-# \d topics  
Table "public.topics"  
Column | Type | Collation | Nullable | Default  
-----+-----+-----+-----+-----  
topic_id | integer | not null | nextval('topics_topic_id_seq')::regclass  
topic_name | character varying(255) | not null |  
topic_level | integer | not null |  
thumbnail | character varying(255) | not null |  
description | character varying(255) | not null | 'Lorem Ipsum is simply dummynd typesetting industry.....'::character varying  
Indexes:  
    "pk_topic" PRIMARY KEY, btree (topic_id)  
Check constraints:  
    "topic_chk_valid_level" CHECK (topic_level > 0 AND topic_level < 4)  
Referenced by:  
    TABLE "posts" CONSTRAINT "fk_posts_topicid" FOREIGN KEY (topic_id) REFERENCES topics(topic_id)  
    TABLE "tests" CONSTRAINT "fk_test_topic_id" FOREIGN KEY (topic_id) REFERENCES topics(topic_id)
```

Hình 13: Topics Schema

Bảng users:

```
trainning-website-# \d users
```



```
tranning-website-# \d users
                                         Table "public.users"
   Column |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----+
  uuid  | integer        |           | not null | nextval('users_uuid_seq'::regclass)
username | character varying(255) |           | not null |
  email  | character varying(100) |           | not null |
  passrd | character varying(255) |           |
created_date | date |           |
  is_active | boolean |           |           | false
  is_admin  | boolean |           |           | false
  u_avatar  | character varying(255) |           |
 public_id | character varying(255) |           |
Indexes:
  "pk_user" PRIMARY KEY, btree (uuid)
  "users_public_id_key" UNIQUE CONSTRAINT, btree (public_id)
  "users_username_key" UNIQUE CONSTRAINT, btree (username)
Check constraints:
  "users_chk_valid_email" CHECK (email::text ~* '[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$'::text)
  "users_chk_valid_username" CHECK (length(username::text) > 5 AND username::text ~> 'admin'::text AND username::text ~> 'username'::text)
Referenced by:
  TABLE "tests" CONSTRAINT "fk_tests_uid" FOREIGN KEY (uuid) REFERENCES users(uuid)
```

Hình 14: Users Schema

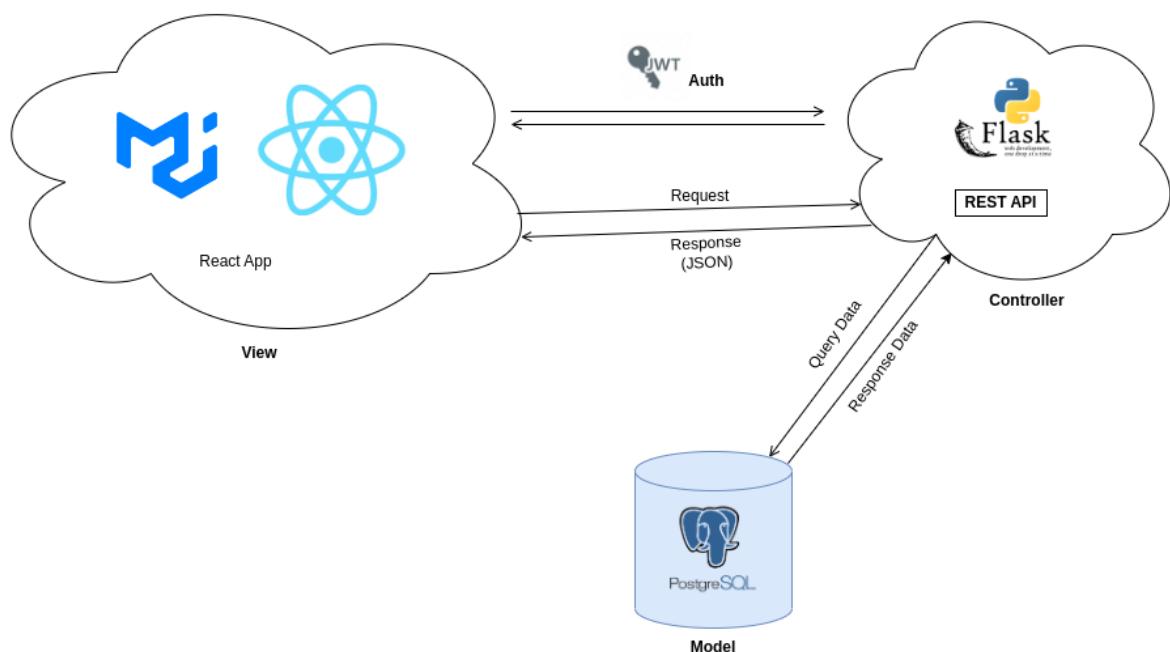
4. Hiện thực

Toàn bộ thông tin về mã nguồn của hệ thống được lưu trữ tại link Github: <https://github.com/thanhdxuan/summer-training>

4.1. Tổng quan kiến trúc hệ thống

Ở đây, sinh viên sử dụng kiến trúc MVC (Model - View - Controller) để thiết kế ứng dụng.

- **Model:** Các model để lưu trữ dữ liệu của hệ thống như **Topics**, **Users**, ...
- **View:** Các module, component sử dụng ReactJS để hiện thực, giúp người dùng thao tác với giao diện trang web, từ đó thực hiện các yêu cầu (request) thông qua các REST API được phía server cung cấp.
- **Controller:** Là các module, routes để xử lý các yêu cầu từ **View**, từ đó thực hiện các logic cần thiết, cũng như query dữ liệu từ **Model** nếu cần thiết sau đó trả kết quả về cho **View** dưới dạng JSON để hiển thị.



Hình 15: Tổng quan kiến trúc hệ thống

Bảo mật

Ở đây, hệ thống sử dụng JWT (JSON Web Token), để xác nhận người dùng trước khi sử dụng hệ thống.

JWT là một chuỗi ký tự có ba phần chính: Header, Payload và Signature. Header chứa các thông tin về loại mã hóa và thuật toán được sử dụng. Payload chứa các thông tin tùy chỉnh được gửi kèm trong token, ví dụ như thông tin người dùng và quyền truy cập. Signature được tạo bằng cách ký (sign) Header và Payload bằng một khóa bí mật duy nhất, để đảm bảo tính xác thực của token. Khóa bí mật này thường được lưu trữ ở phía server.

Quá trình hoạt động của JWT như sau:

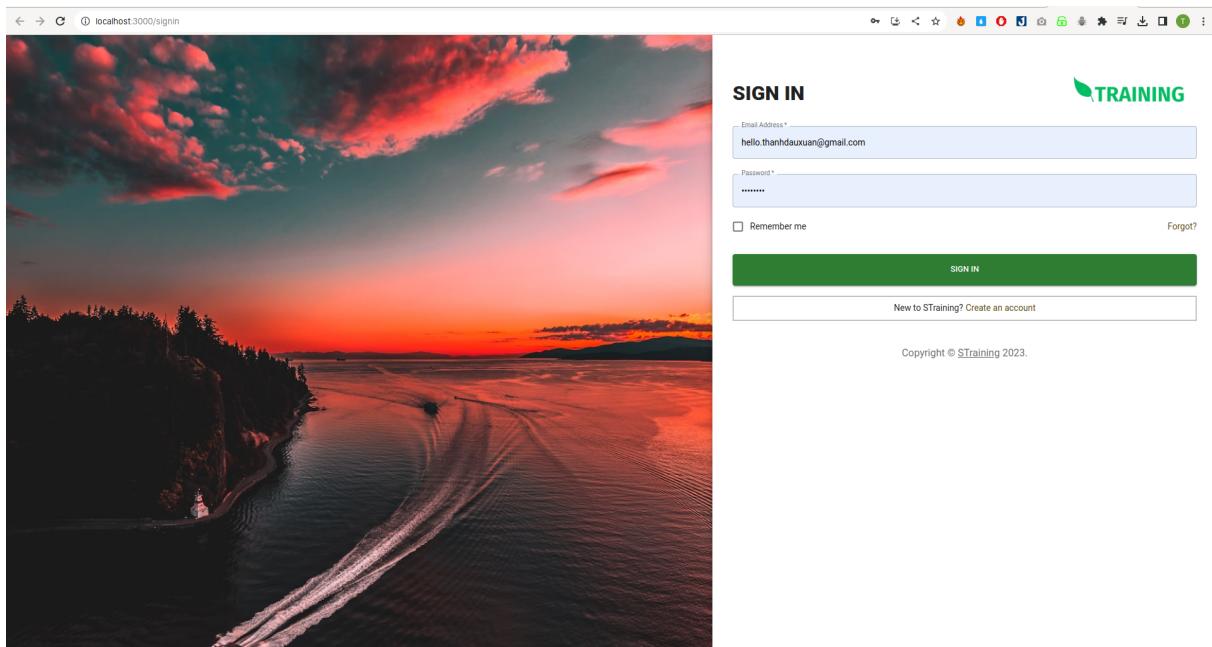
- Người dùng gửi yêu cầu xác thực đến máy chủ.
- Máy chủ xác thực thông tin người dùng và tạo một JWT.
- JWT được gửi trả lại cho người dùng.
- Người dùng gửi JWT trong mỗi yêu cầu tiếp theo.

- Máy chủ xác thực JWT bằng cách kiểm tra tính hợp lệ, xác minh chữ ký và kiểm tra quyền truy cập.
- Nếu JWT hợp lệ, máy chủ trả về kết quả yêu cầu cho người dùng.

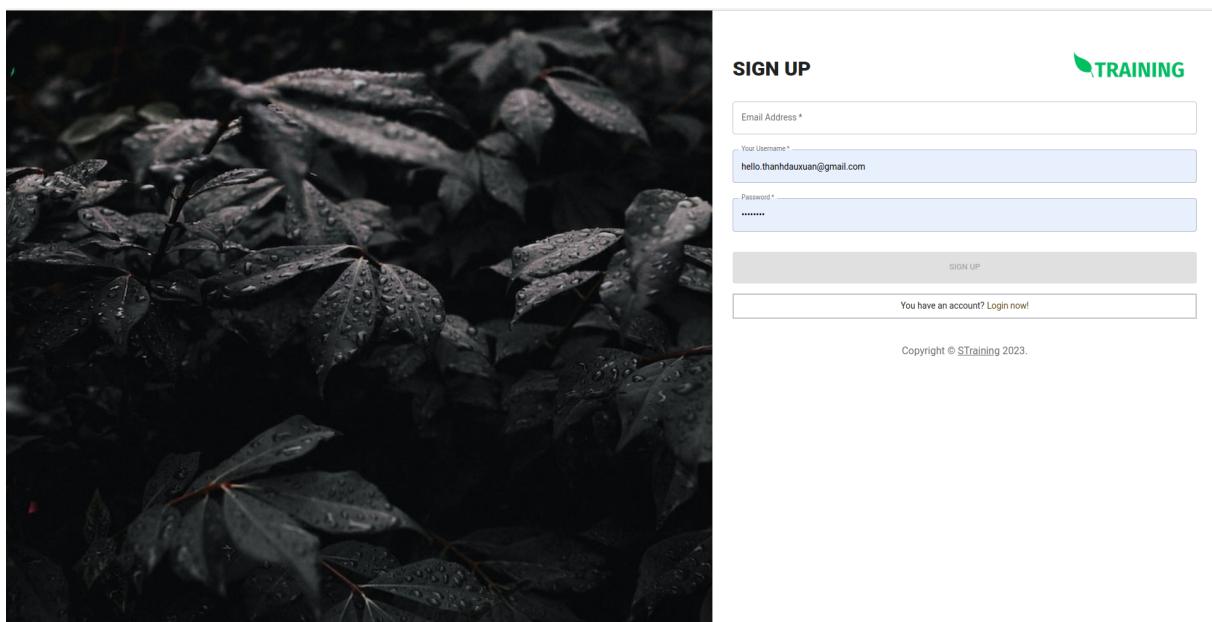
Khi người dùng đăng nhập trang web, trang web sẽ xác thực và trả về JWT cho phía client, phía client lưu trữ token này để thực hiện các yêu cầu sau đó.

4.2. Giao diện

4.2.1. Đăng nhập, đăng ký



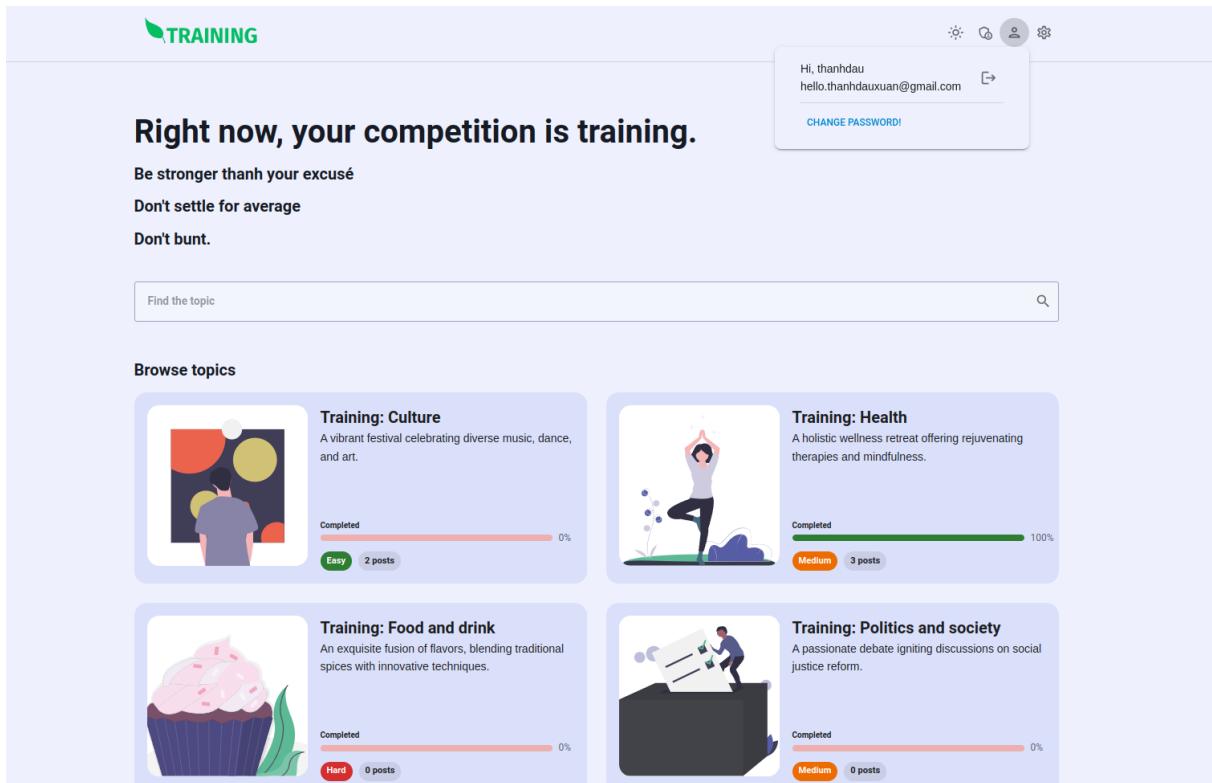
Hình 16: Giao diện khi đăng nhập



Hình 17: Giao diện khi đăng ký

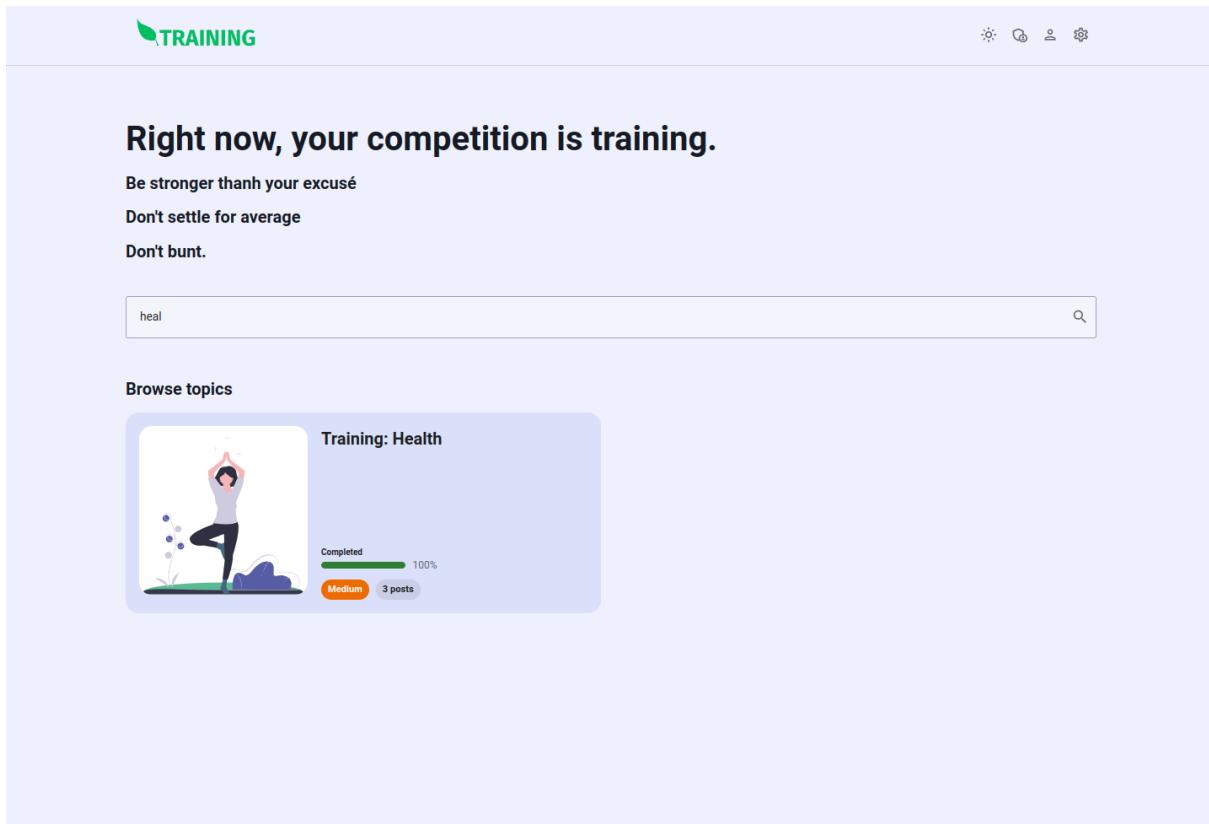
4.2.2. Trang chủ

Khi người dùng đăng nhập thành công vào hệ thống, mới có thể xem được các chủ đề hiện có, và số chủ đề đã hoàn thành / chưa hoàn thành.



Hình 18: Trang chủ

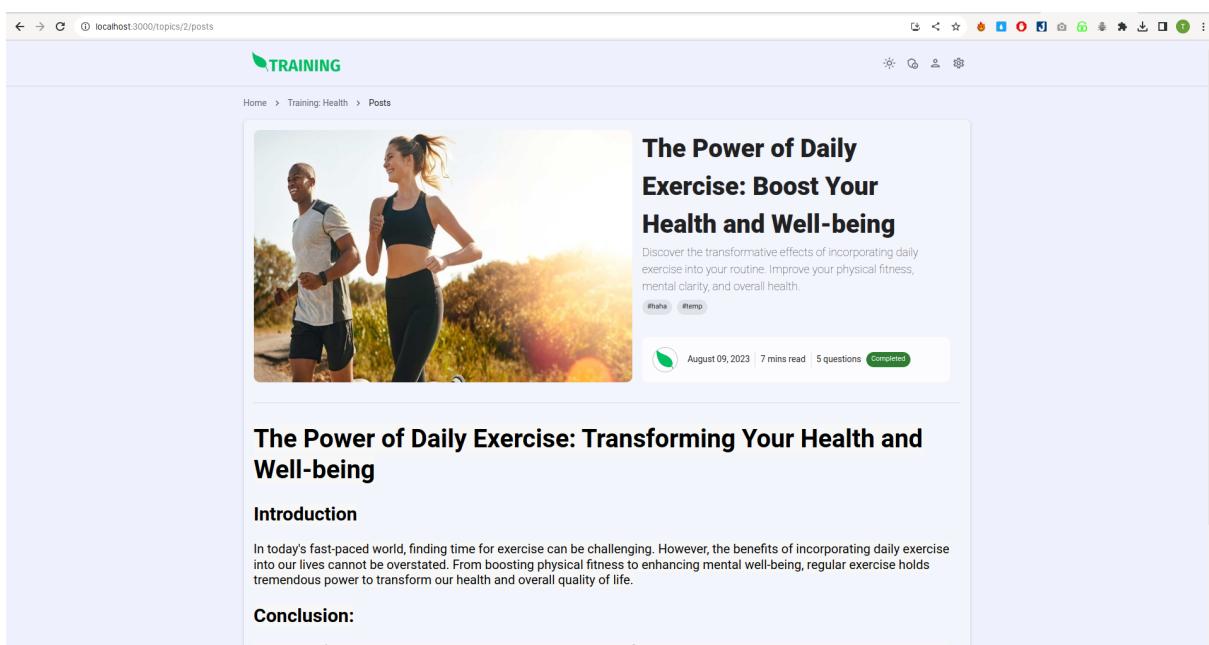
Tại trang chủ, người dùng có thể tìm kiếm chủ đề dựa trên tên của chủ đề đó.



Hình 19: Tìm kiếm chủ đề

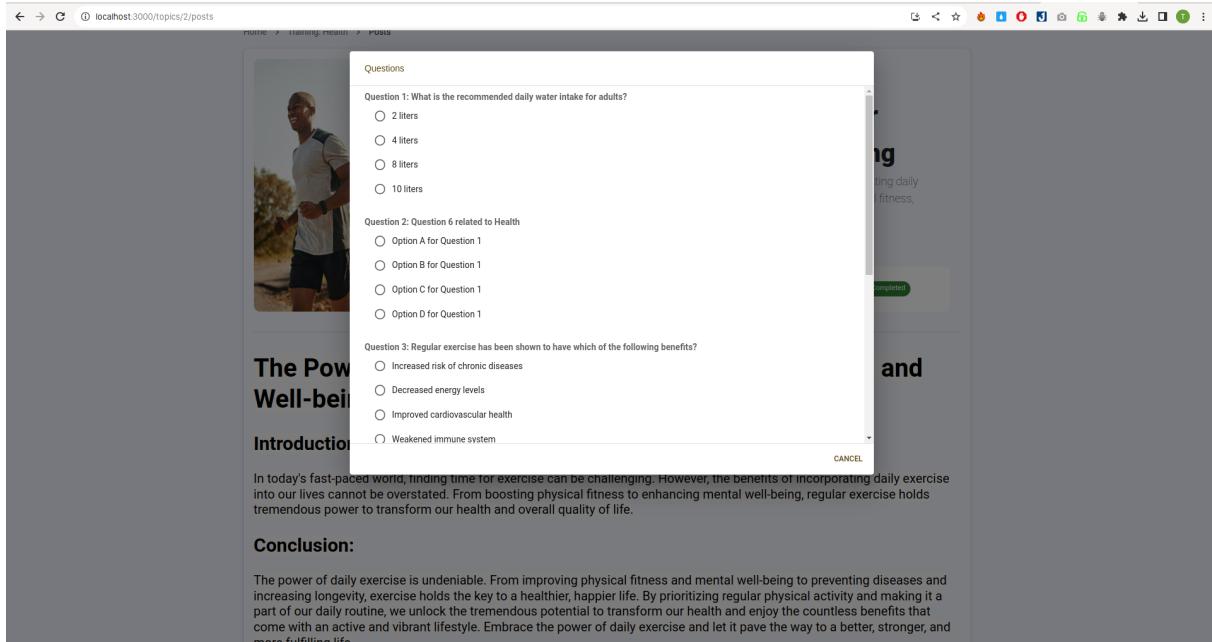
4.2.3. Bài đăng

Khi người dùng chọn vào chủ đề bất kỳ, hệ thống sẽ hiển thị các bài đăng của chủ đề đó, theo thứ tự.



Hình 20: Mỗi bài đăng được Admin tạo bằng html editor

Ở mỗi bài đăng, hệ thống yêu cầu người dùng phải trả lời câu hỏi cho bài đăng đó, bằng cách nhấn vào button ở cuối trang.



Hình 21: Dialog hiển thị các câu hỏi dưới dạng trắc nghiệm

4.2.4. Admin Panel

Với người dùng là tài khoản admin, sẽ có một trang riêng biệt để có thể xem các thông tin của người dùng, các bài đăng, chủ đề hiện có, từ đó có thể thực hiện các thao tác như thêm, xóa, sửa.

Users

Admin có thể xem các thông tin cơ bản của tất cả các người dùng đang có trong hệ thống, xem loại tài khoản, trạng thái (**đã được kích hoạt hay chưa**), từ đó có thể kích hoạt tài khoản hoặc hủy kích hoạt.

Hình 22: Quản lý người dùng

Chủ đề

Admin có thể xem các chủ đề hiện có, hoặc tạo thêm chủ đề mới.



ID	Name	Number of Posts	Level	Delete
1	Training: Culture	2	Easy	
2	Training: Health	3	Medium	
3	Training: Food and drink	0	Hard	
4	Training: Politics and society	0	Medium	
16	Training: Zhao Lusi	0	Easy	

Rows per page: 5 ▾ 1–5 of 5 < >

[CREATE NEW TOPIC](#)

Hình 23: Xem thông tin các chủ đề hiện có

Topic name *

Description *

UPLOAD THUMBNAIL
Please upload the topic thumbnail

[CREATE TOPIC](#)

Hình 24: Tạo chủ đề mới

Bài đăng

Admin có thể xem thông tin của các bài đăng hiện có của một chủ đề cụ thể, hoặc tạo thêm / chỉnh sửa bài đăng cho chủ đề đó.

Light Mode | Dark Mode | User | Settings

Home > Admin

USERS TOPICS POSTS QUESTIONS

Select the topic
Select...
Training: Health

ID	Title	Number of questions	Created Date	Delete	Edit
1	The Power of Daily Exercise: Boost Your Health and Well-being	5	August 09, 2023		
2	Nutrition Essentials: Nourish Your Body for Optimal Health and Vitality	3	August 10, 2023		
3	Stress Management Techniques: Achieve Inner Balance and Mental Well-being	5	August 11, 2023		

CREATE NEW POST

Hình 25: Xem thông tin bài đăng của một chủ đề

Post title* _____

What is ER Model?

Estimated Read Time* _____

4

Description*

Normal ♦ B I U ☰ ≡ ≡ Tx

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore

et dolore magnam aliquam quaerat voluptatem. Ut enim aequo doleamus animo, cum corpore
dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum
nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distingue

1. possit augeri amplificari que non possit. At etiam Athenis, ut e patre audiebam facete et urbane
 Stoicos irridente, status est in qua a nobis philosophia defensa et collaudata est, cum id, quod
 maxime placeat facere possimus. omnis voluptas assumenda est, omnis dolor repellendus.

4. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et
 voluptates repudiandas sint et molestiae non recusandas. Itaque earum rerum defuturum, quas
 natura non depravata desiderat. Et quem ad me accedit, saluto: 'chaere' inquam, 'Title' lictores,
 turma omnis chorusque: 'chaere, Title' hinc hostis mi Abucus, hinc inimicus. Sed iure Mucius. Ego
 • autem mirari satis non queo under hoc sit tam insolens domesticarum rerum fastidium. Non est
 • omnino hic docendi locus, sed ita prorsus existimo, neque eum Torquatum, qui hoc primus
 • cognomen inveniet, aut torquem illum hosti detraxisse, ut aliquam ex eo est consecutus? – Laudem
 • et caritatem, quae sunt vitae.

UPLOAD BANNER

Please upload the banner for this post.

SUBMIT

Hình 26: Tao bài đăng cho một chủ đề



The screenshot shows a 'Edit post' modal over a dark background. The modal has a title 'Edit post' and fields for 'Post title*' (The Power of Daily Exercise: Boost Your Health and Well-being), 'Estimated Read Time*', and 'Description'. The description text is: 'Discover the transformative effects of incorporating daily exercise into your routine. Improve your physical fitness, mental clarity, and overall health.' Below the description is a rich text editor toolbar. The main content area contains two sections: 'Introduction' and 'Conclusion'. The introduction section includes a paragraph about the benefits of daily exercise. The conclusion section includes a paragraph encouraging readers to embrace daily exercise. At the bottom right of the modal are 'CANCEL' and 'UPDATE' buttons.

Hình 27: Chỉnh sửa bài đăng

Câu hỏi

Tại đây, admin có thể quản lý thông tin về câu hỏi của một bài đăng, có thể tạo/xóa câu hỏi.

The screenshot shows the 'QUESTIONS' tab of a management interface. It displays a list of questions for a post titled 'The Power of Daily Exercise: Boost Your Health and Well-being'. The questions are listed in a table with columns for ID, Title, Options, Answer, and Delete. The first question is 'What is the recommended daily water intake for adults?' with options A. 2 liters, B. 4 liters, C. 8 liters, D. 10 liters. The second question is 'Question 6 related to Health' with options A. Option A for Question 1, B. Option B for Question 1, C. Option C for Question 1, D. Option D for Question 1. The third question is 'Regular exercise has been shown to have which of the following benefits?' with options A. Increased risk of chronic diseases, B. Decreased energy levels, C. Improved cardiovascular health, D. Weakened immune system. The fourth question is 'What is the recommended duration of moderate-intensity aerobic exercise per week for adults, according to health guidelines?' with options A. 15 minutes, B. 30 minutes, C. 60 minutes, D. 120 minutes. The fifth question is 'Resistance training exercises, such as weightlifting, help to:' with options A. Reduce flexibility, B. Increase the risk of injuries, C. Build muscle strength and mass, D. Decrease bone density. At the bottom right of the table are buttons for 'Rows per page' (5), '1–5 of 5', and 'CREATE NEW QUESTION'.

Hình 28: Xem thông tin câu hỏi



CREATE NEW QUESTION

Question *

Options 1 *

Options 2 *

Options 3 *

Options 4 *

Answer (1 - 4) *

SUBMIT

Hình 29: Tạo câu hỏi mới



Tài liệu tham khảo

- [1] MUI, “Material ui,” 2019. [Online]. Available: <https://mui.com/material-ui>
- [2] P. Tutorial, “What is postgresql?,” 2015. [Online]. Available: <https://www.postgresqltutorial.com/postgresql-getting-started/what-is-postgresql/>
- [3] F. Authors, “Flask documentation,” 2019. [Online]. Available: <https://flask.palletsprojects.com/en/2.3.x/>