# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

- Trong quá trình thực hiện đề tài "Xây dựng website đăng ký đề tài và giám sát thực hiện khóa luận tốt nghiệp cho sinh viên Khoa CNTT - IUH", các phần nội dung lý thuyết sau đây được vận dụng:

## **2.1 Kiến thức về phát triển web:**

### **HTML**



1. Tổng quan

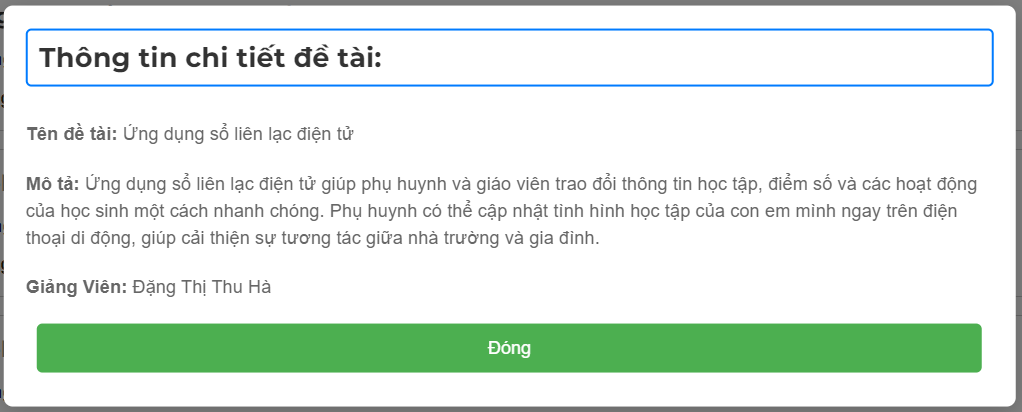
* HTML được sử dụng để xây dựng cấu trúc của trang web, HTML được sử dụng thông qua JSX, là một cú pháp mở rộng của JavaScript trong React để viết cấu trúc giao diện người dùng.

1. Hiện thực code



*Hình 2.1.1.1 Mô tả code HTML trong code thực tế*

* **<div>**, **<h3>**, **<p>**, **<strong>**, **<button> :**  Là các thẻ HTML để cấu trúc giao diện hiển thị.
* Nội dung như **Tên đề tài**, **Mô tả**, và **Giảng Viên** được hiển thị trong các thẻ **<p>**, kèm theo **<strong>** để làm nổi bật tiêu đề.
* **<button>** được sử dụng để tạo nút "Đóng".



*Hình 2.1.1.2 Mô giao diện của code*

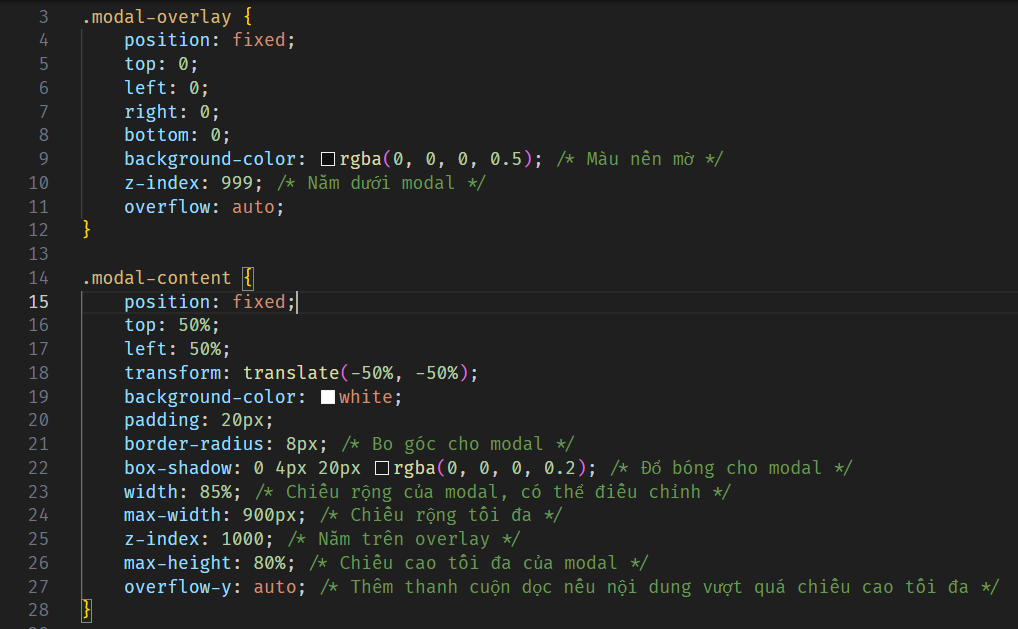
### **CSS**



1. Tổng quan

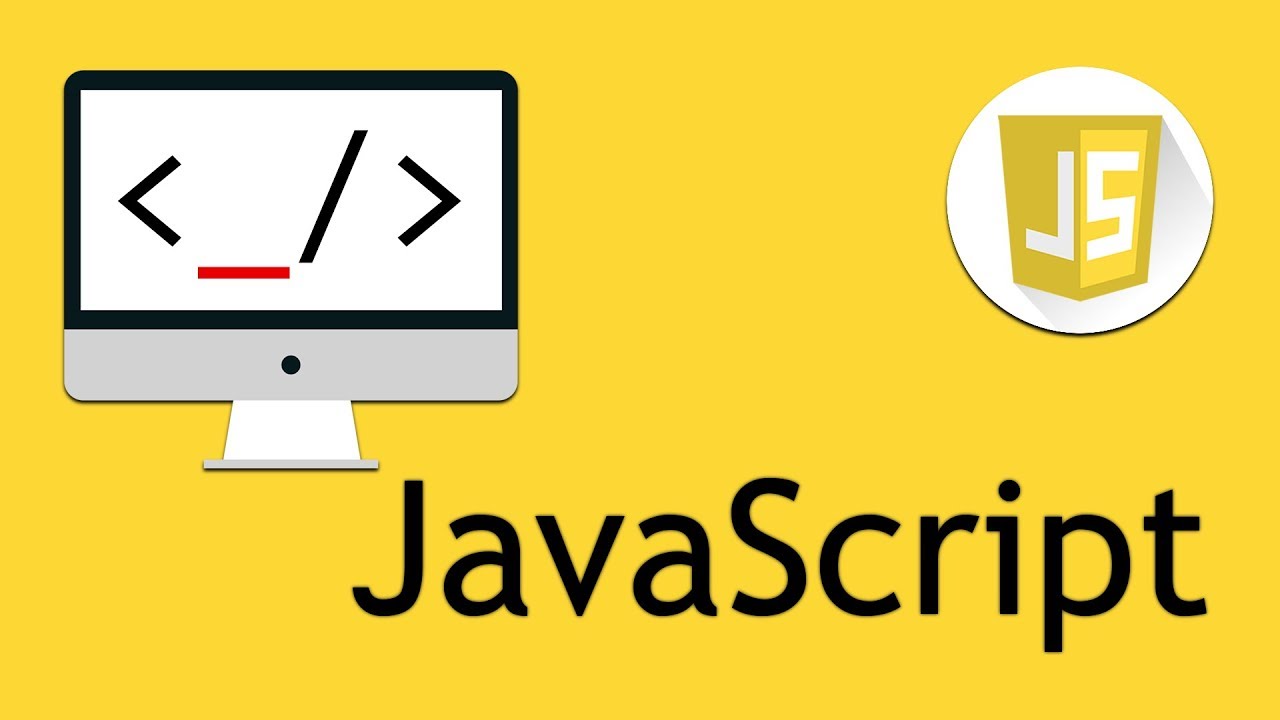
* CSS (Cascading Style Sheets):được sử dụng để định dạng và thiết kế giao diện của trang web, làm cho nó trực quan và thân thiện với người dùng.

1. Hiện thực code



* **Border-radius: 8px;** Dùng để **bo tròn các góc** của phần tử. Ở đây, border-radius: 8px; có nghĩa là các góc của modal sẽ được bo tròn với bán kính là 8px. Nó giúp modal trông mềm mại và không bị cứng nhắc với các góc vuông.
* **Width: 85%;** Xác định **chiều rộng** của phần tử. Ở đây, **width: 85%;** có nghĩa là modal sẽ chiếm 85% chiều rộng của màn hình (hoặc phần tử chứa nó). Điều này giúp modal có kích thước tương đối linh hoạt, điều chỉnh theo kích thước màn hình của người dùng.

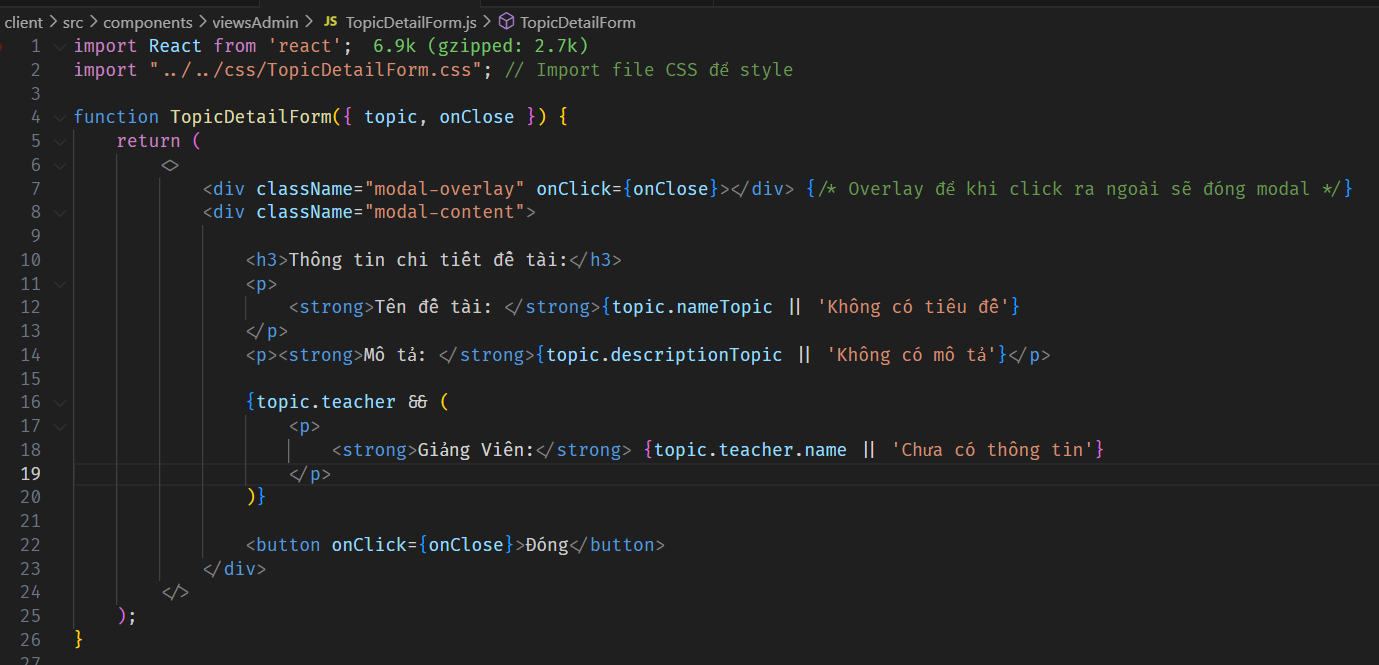
### **JavaScript**



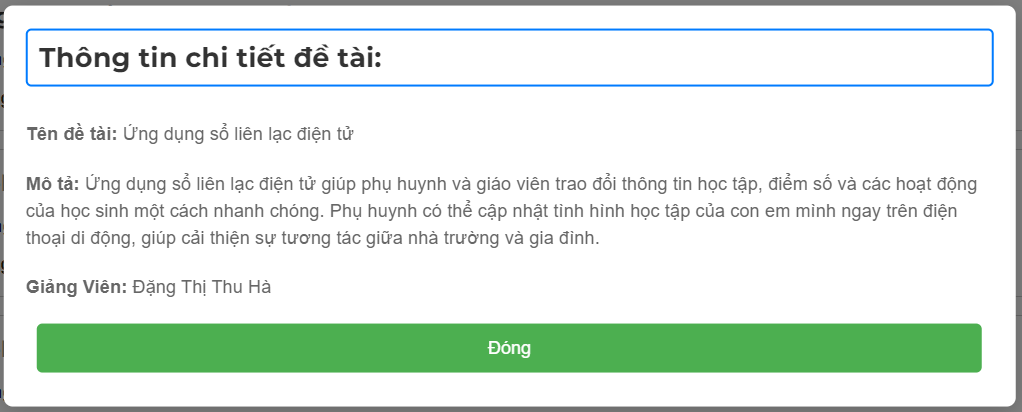
1. Tổng quan

* JavaScript:là ngôn ngữ lập trình phía client giúp thêm tính năng động và xử lý sự kiện trên trang web. Đoạn mã React phía dưới chính là JavaScript kết hợp với JSX để tạo ra thành phần động cho modal.

1. Chi tiết code

****

* **import React:** Sử dụng React để tạo thành phần giao diện động.
* **TopicDetailForm function:** Đây là một thành phần React hiển thị modal chứa thông tin đề tài, giảng viên. Hàm này nhận hai tham số **topic** (dữ liệu đề tài) và **onClose** (hàm đóng modal).
* **onClick={onClose}:** Đây là xử lý sự kiện trong **JavaScript** để đóng **modal** khi người dùng click vào **overlay** hoặc nút "Đóng".
* **Điều kiện hiển thị: {topic.teacher && (...)}:** Là cú pháp **JavaScript** giúp kiểm tra xem có thông tin giảng viên hay không, nếu có thì sẽ hiển thị.

****

### **2.1.4 Responsive Design**

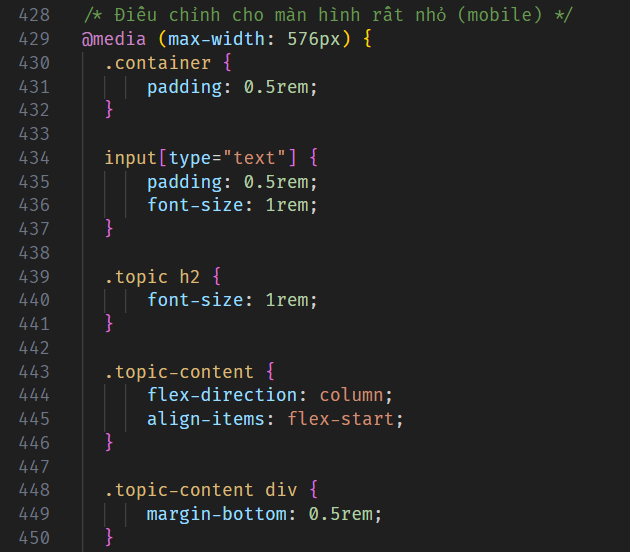


1. Tổng quan

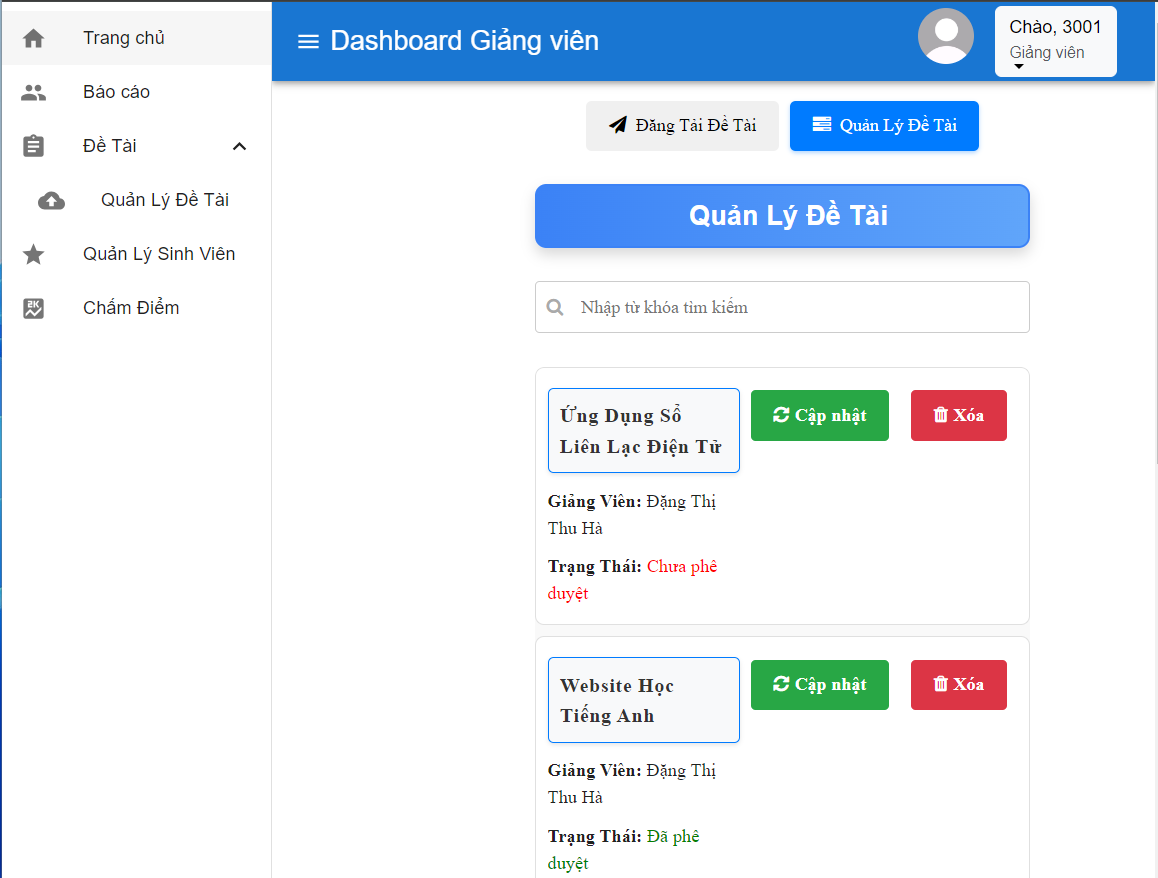
* Responsive Design: Kỹ thuật này được sử dụng để đảm bảo rằng giao diện của website có thể thích ứng tốt với các kích thước màn hình khác nhau, từ máy tính để bàn đến điện thoại di động.

1. Hiện thực code



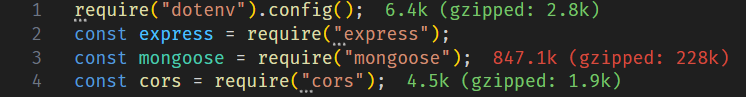


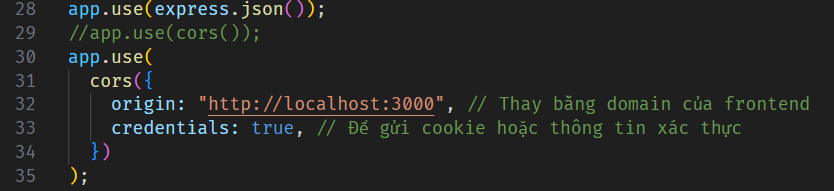
**@media (max-width: 768px)** và **@media (max-width: 576px)** được sử dụng để kiểm tra kích thước màn hình và áp dụng các style phù hợp khi chiều rộng của màn hình nhỏ hơn 768px hoặc 576px (thường là tablet và mobile).

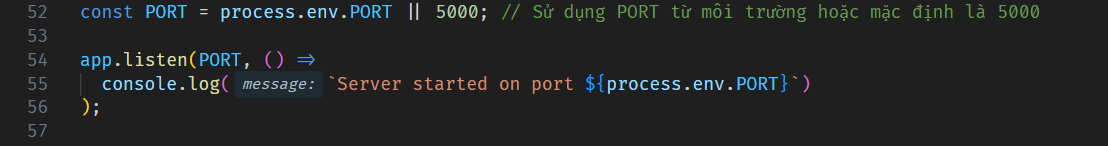


## **2.2 Frameworks và thư viện phát triển web:**

* **Node.js** là một môi trường chạy JavaScript trên server, giúp bạn xây dựng các ứng dụng mạng hiệu quả và linh hoạt. Với Node.js, bạn có thể thực hiện nhiều thao tác xử lý đồng thời mà không bị chậm nhờ vào cơ chế xử lý không đồng bộ của nó.

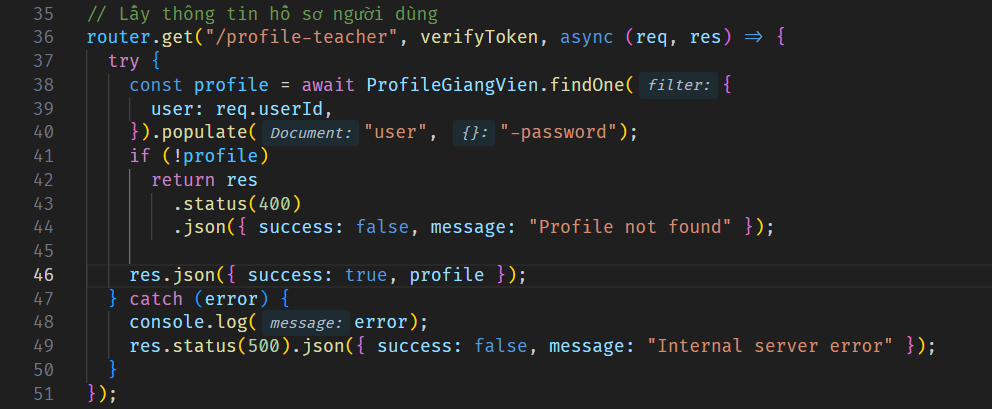




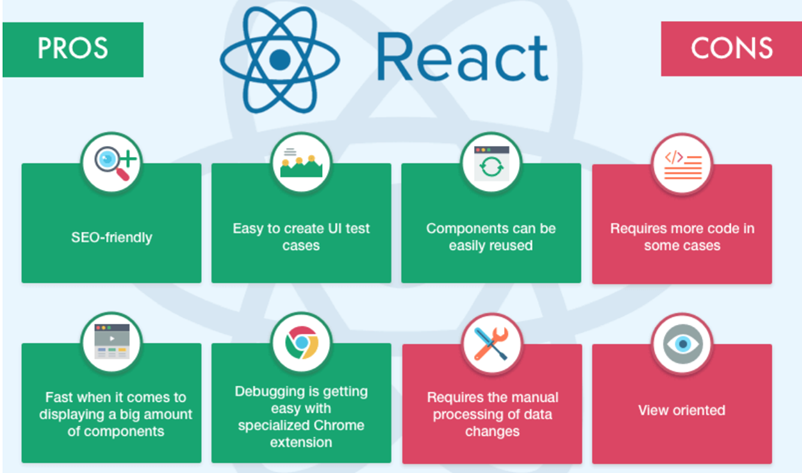


Node.js tạo ra một server và lắng nghe trên cổng 5000.

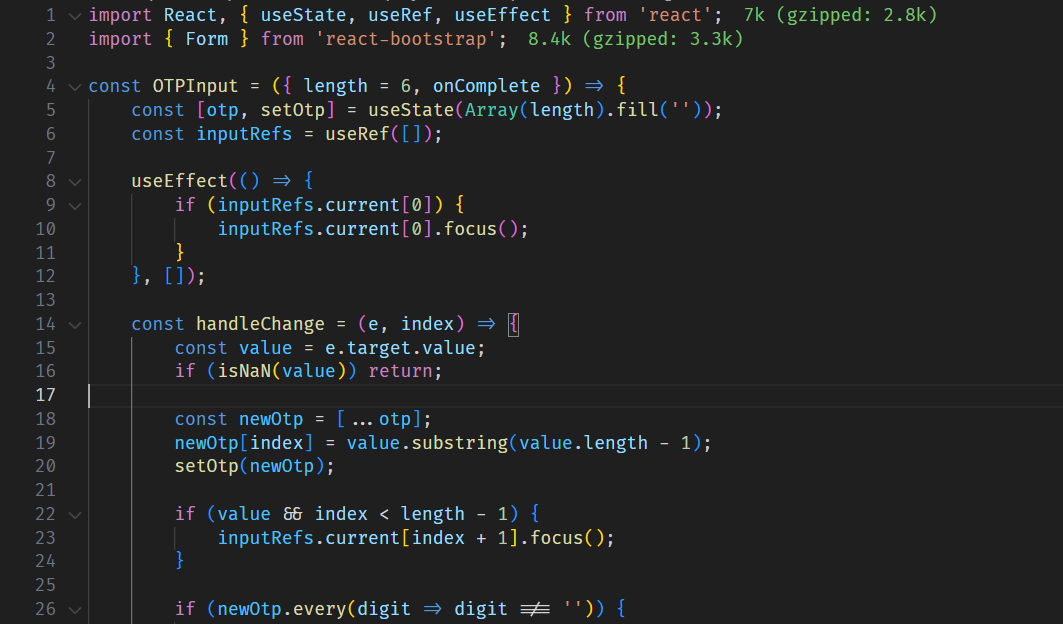
* **Express.js** là một framework của Node.js, giúp bạn xây dựng các ứng dụng web và API một cách dễ dàng và linh hoạt hơn. Express cung cấp một lớp trừu tượng trên **http** module của Node.js, giúp việc quản lý route, middleware và phản hồi dễ dàng hơn.

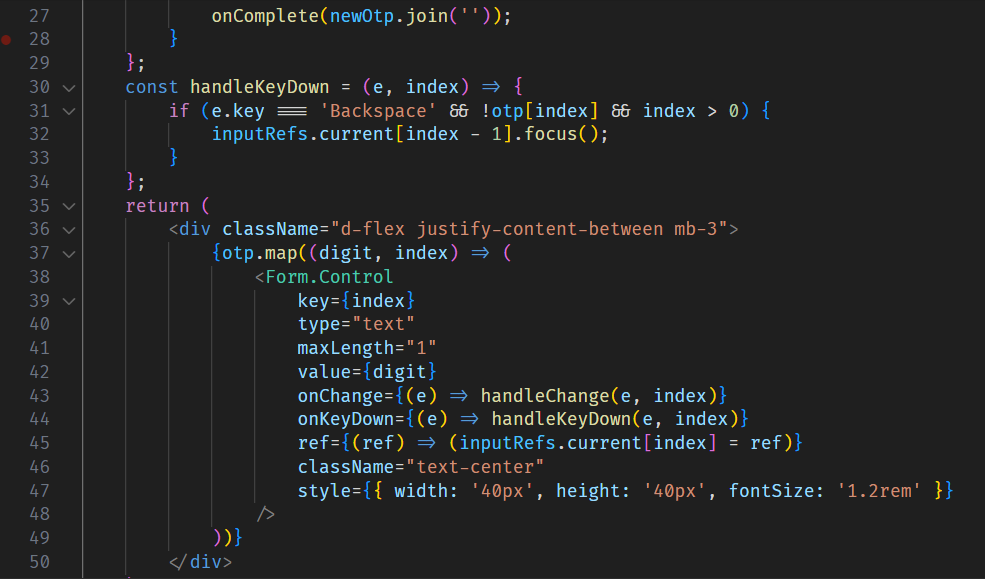


**Express.js** giúp bạn dễ dàng định nghĩa route **/profile-teacher** với phương thức **GET**, và sử dụng middleware **verifyToken** để bảo vệ route này. Express.js còn giúp xử lý lỗi và phản hồi thông qua các phương thức như **res.json()** hoặc **res.status()**.

React:  


* **React.js:** Cho phép bạn tạo ra những phần giao diện của trang web (như nút bấm, hộp thoại, danh sách, v.v.) theo cách linh hoạt và dễ bảo trì hơn. Nó chia trang web thành các thành phần nhỏ (gọi là component), mỗi thành phần có thể tái sử dụng và hoạt động độc lập. ReactJS giúp việc cập nhật và hiển thị giao diện trở nên nhanh chóng và mượt mà hơn, đặc biệt khi có nhiều dữ liệu thay đổi.

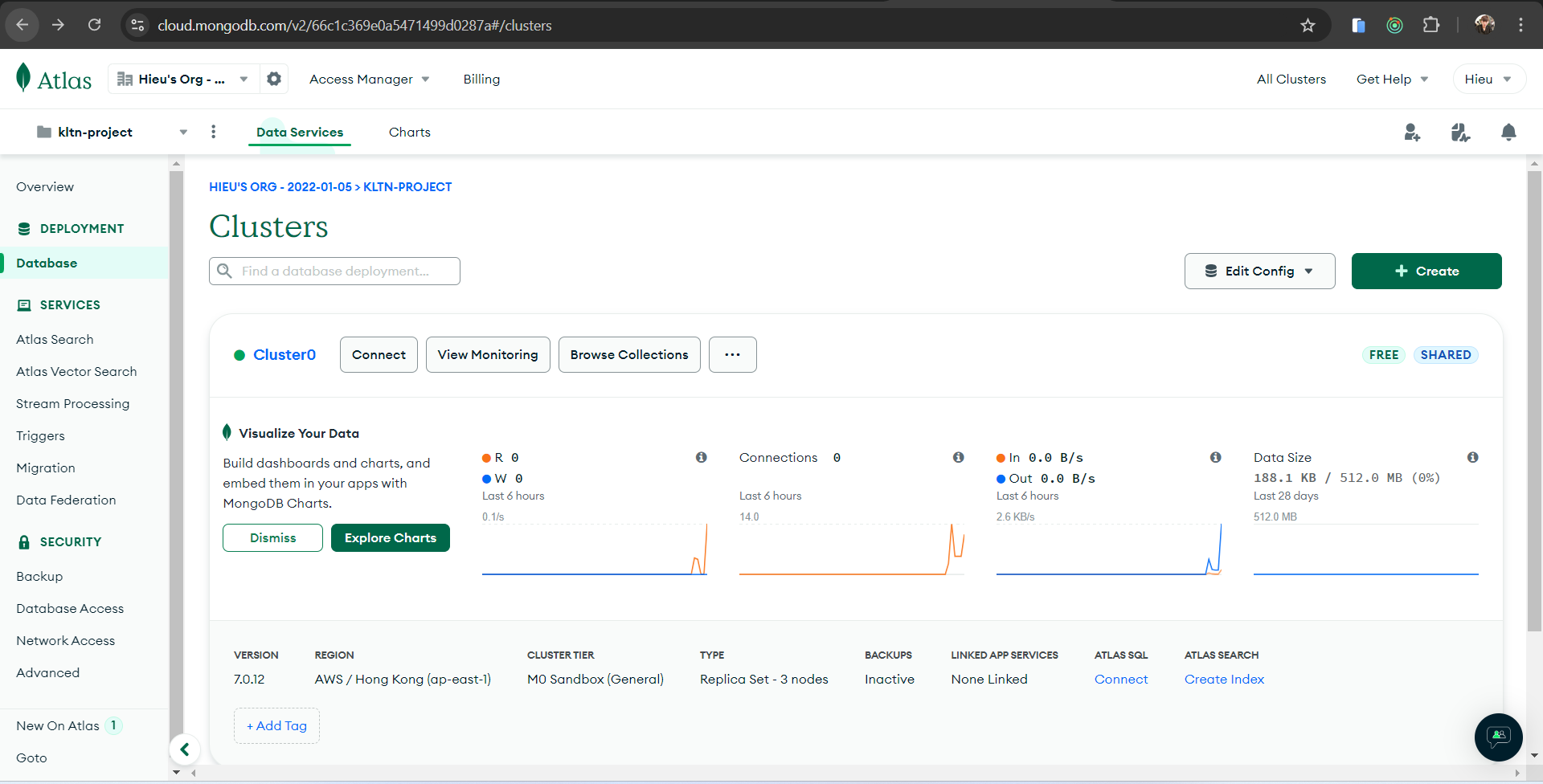


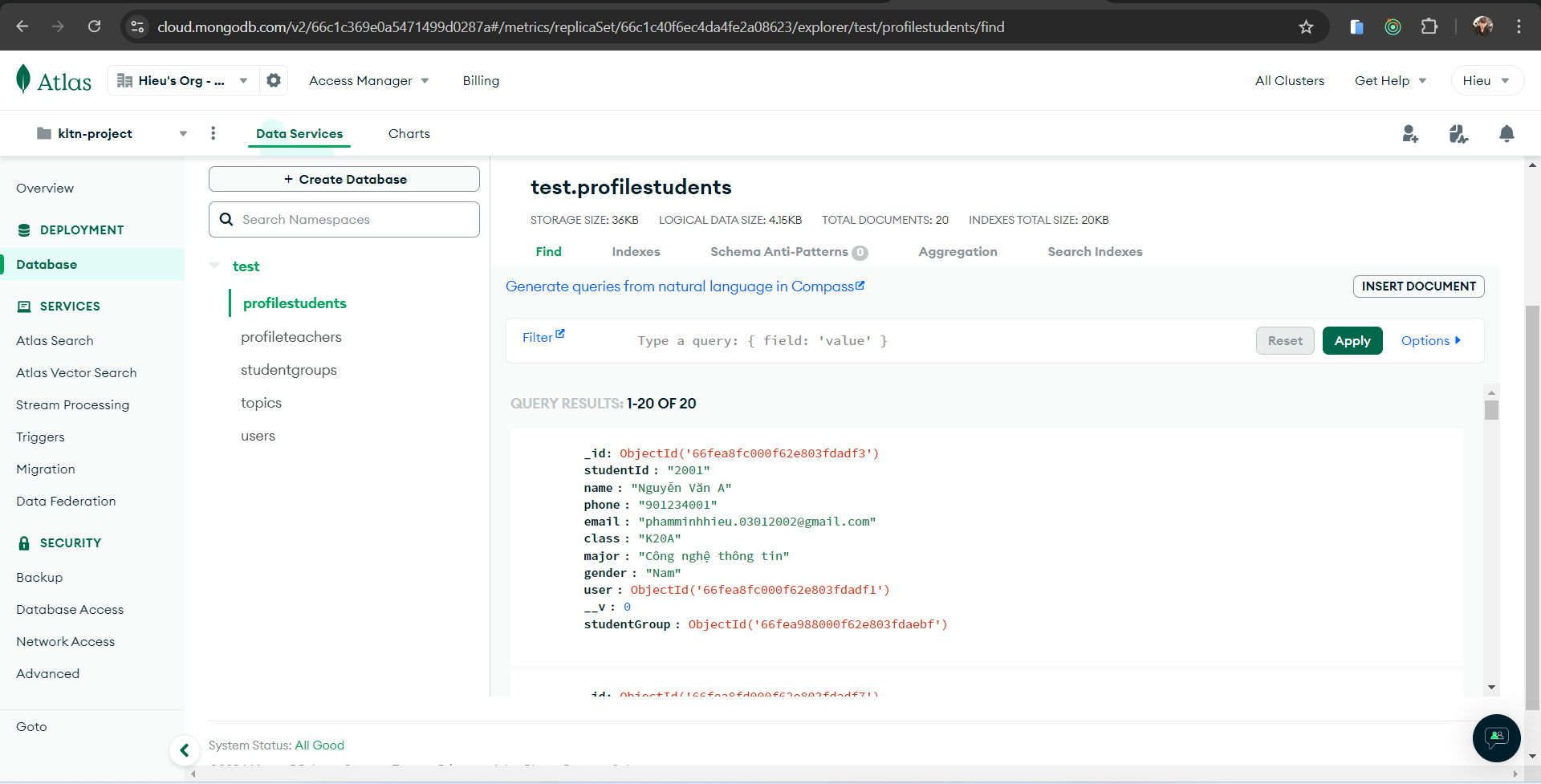


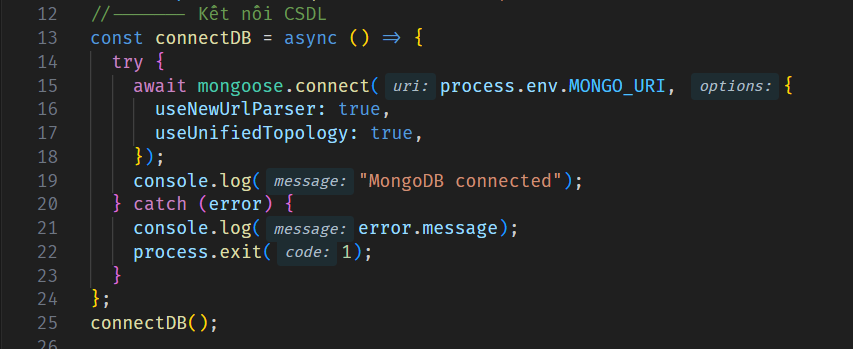


## **2.3 Quản lý cơ sở dữ liệu:**

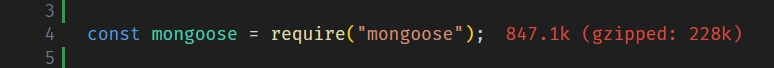
* **MongoDB:** Đây là hệ quản trị cơ sở dữ liệu NoSQL, được sử dụng để lưu trữ và quản lý dữ liệu liên quan đến các đề tài khóa luận, người dùng, tiến độ thực hiện, và các thông tin khác. MongoDB cho phép lưu trữ dữ liệu dưới dạng JSON, giúp cho việc truy vấn và thao tác dữ liệu trở nên linh hoạt và nhanh chóng.

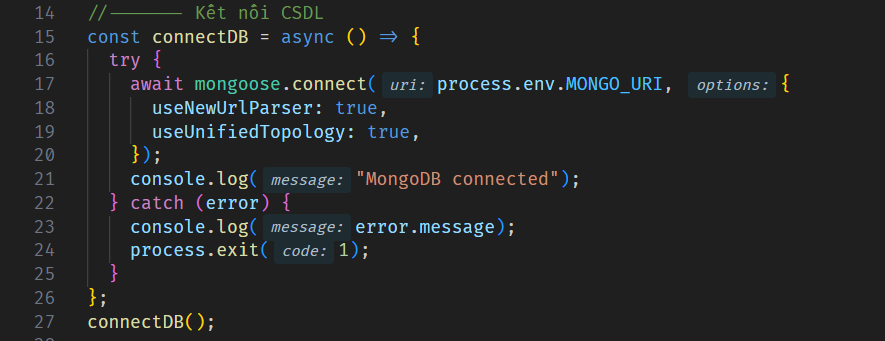






* **Mongoose:** Một ODM (Object Data Modeling) library cho MongoDB và Node.js, giúp việc tương tác với cơ sở dữ liệu MongoDB trở nên dễ dàng hơn thông qua các mô hình dữ liệu, và cung cấp các tính năng quản lý schema và validation.

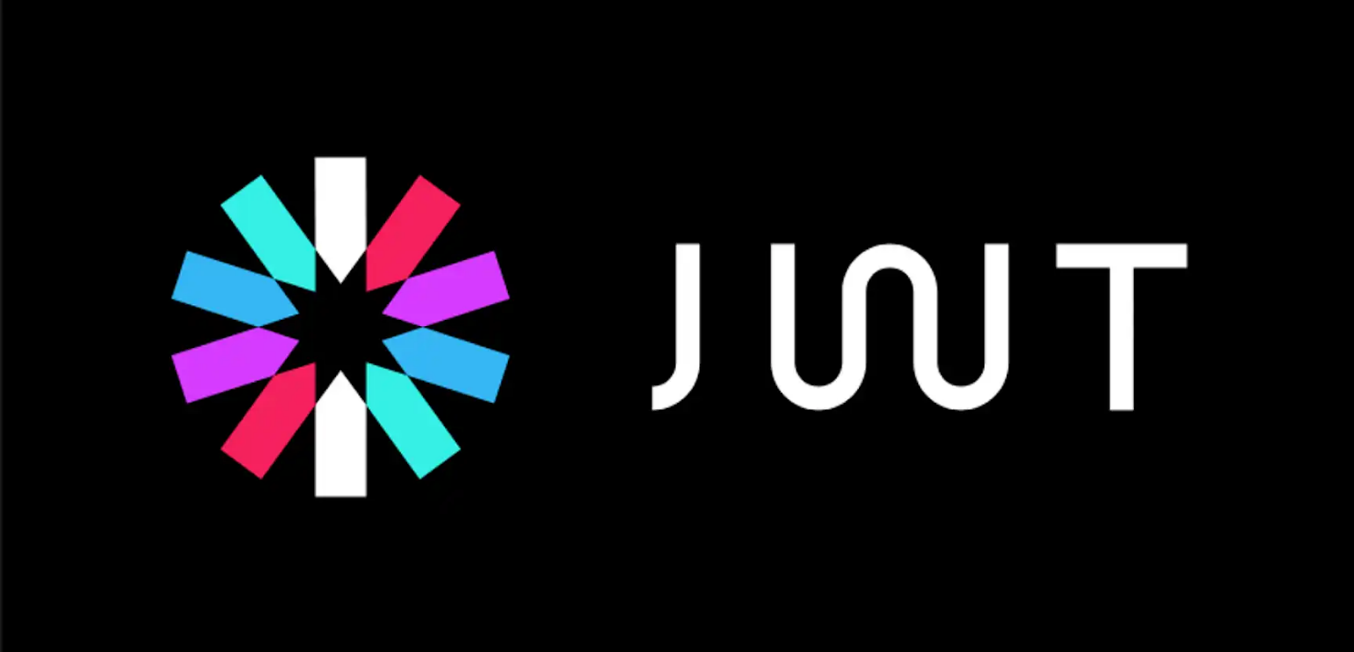


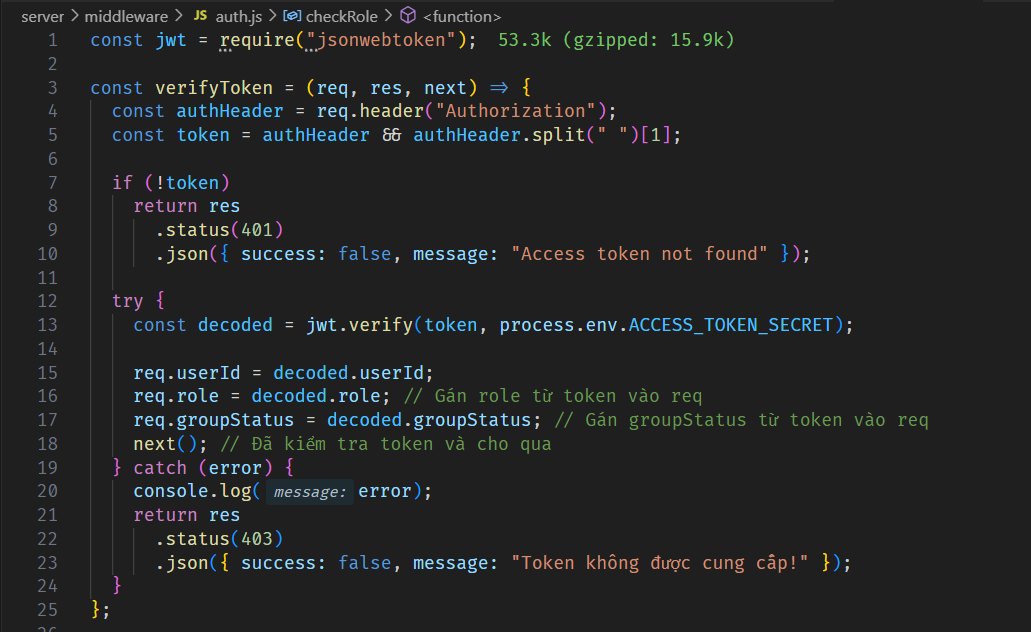


## **2.4 An ninh mạng và bảo mật thông tin:**

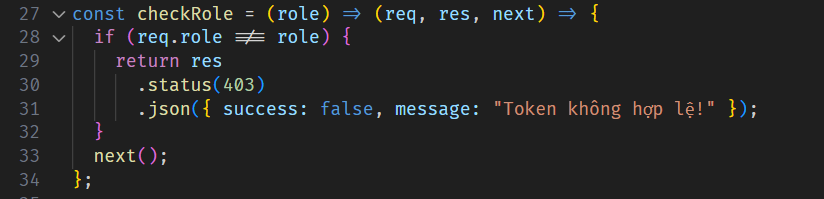
* **Xác thực và phân quyền:** Đảm bảo rằng chỉ những người dùng được ủy quyền mới có thể truy cập vào các phần cụ thể của hệ thống. Điều này được thực hiện thông qua các phương pháp xác thực như JWT (JSON Web Token) và phân quyền người dùng.

+ Xác thực JWT (JSON Web Token)

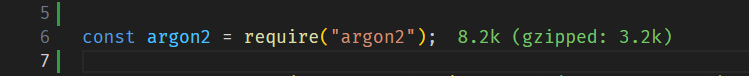




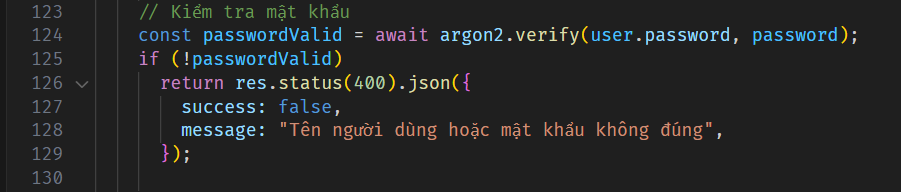
+ Phân quyền



* **Mã hóa dữ liệu:** Sử dụng các kỹ thuật mã hóa để bảo vệ dữ liệu nhạy cảm trong quá trình lưu trữ và truyền tải.



+ Mã hóa dữ liệu bằng **argon2**: là một thuật toán mã hóa (hashing) mạnh mẽ được thiết kế để bảo vệ mật khẩu và các thông tin nhạy cảm khác. Nó được phát triển vào năm 2015 và đã giành được giải thưởng **Password Hashing Competition (PHC)**, xác nhận tính hiệu quả và bảo mật của nó.



## **2.5 Triển khai và quản trị hệ thống:**

* **Hosting và triển khai:** Sử dụng các dịch vụ như Heroku, AWS, Netlify, Render, hoặc DigitalOcean để triển khai và quản lý website trên môi trường thực tế.
* **Quản lý phiên bản:** Sử dụng Git và GitHub để quản lý mã nguồn, theo dõi các thay đổi, và hợp tác với các thành viên khác trong nhóm phát triển.

## **2.6 Nghiên cứu tài liệu và tự học:**

* **Tìm hiểu tài liệu:** Tự tìm hiểu và áp dụng các công nghệ mới, các framework hiện đại hoặc các phương pháp tốt nhất trong phát triển web và quản lý hệ thống, dựa trên tài liệu, sách vở và các khóa học trực tuyến.
* **Áp dụng kiến thức mới:** Áp dụng các kiến thức tự học để giải quyết các vấn đề phức tạp hoặc cải tiến hiệu suất của hệ thống, góp phần vào sự hoàn thiện của đề tài.