|  |
| --- |
| **TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**  **KHOA CÔNG NGHỆ THÔNG TIN**        **BÁO CÁO CUỐI KỲ**  **ĐỀ TÀI: HỆ THỐNG QUẢN LÝ HỌC TẬP SINH VIÊN**    **Môn học: Lập trình cơ sở dữ liệu**  **GVHD: ThS. Phạm Chí Công**  **LỚP: (KI 162-CS2101)**  **SINH VIÊN THỰC HIỆN:**  **Y Jhon ÊBan - 2151010458**  **Nguyễn Văn Mau – 2151010218**  **Nguyễn Trần Đức Anh - 2151010013**      **Thành phố Hồ Chí Minh, 2024** |

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN 3](#_Toc166859483)

[1.Giới thiệu: 3](#_Toc166859484)

[1.1 Giới thiệu về đề tài 3](#_Toc166859485)

[1.2 Giới thiệu 1 số chức năng: 3](#_Toc166859486)

[2. Lý do chọn đề tài: 3](#_Toc166859487)

[3. Yêu cầu đề tài: 3](#_Toc166859488)

[CHƯƠNG 2: CÁC CÔNG NGHỆ ĐƯỢC SỬ DỤNG. 4](#_Toc166859489)

[2.1. Các tool được sử dụng: 4](#_Toc166859490)

[2.2. Các công nghệ được sử dụng 4](#_Toc166859491)

[2.3. Cài đặt: 5](#_Toc166859492)

[2.4. Cấu trúc của hệ thống sử dụng mô hình 3 lớp, gồm DAL, BLL, DTO 5](#_Toc166859493)

[1. Presentation Layer (GUI): 6](#_Toc166859494)

[2. Bussiness Layer (BLL) : 7](#_Toc166859495)

[3. Data Layer (DAL) : 7](#_Toc166859496)

[2.5. Database: 7](#_Toc166859497)

[Tổng Quan 8](#_Toc166859498)

[Các Bảng và Mối Quan Hệ 8](#_Toc166859499)

[Bảng Users (Người Dùng) 8](#_Toc166859500)

[Bảng KhoaHoc (Khóa Học) 8](#_Toc166859501)

[Bảng HocKy (Học Kỳ) 9](#_Toc166859502)

[Bảng monhoc (Môn Học) 9](#_Toc166859503)

[Bảng baitap (Bài Tập) 9](#_Toc166859504)

[Bảng loaibaitap (Loại Bài Tập) 9](#_Toc166859505)

[Bảng cauhoi (Câu Hỏi) 9](#_Toc166859506)

[Bảng loaicauhoi (Loại Câu Hỏi) 10](#_Toc166859507)

[Bảng dapan (Đáp Án) 10](#_Toc166859508)

[Bảng ListTracNghiem (Danh Sách Trắc Nghiệm) 10](#_Toc166859509)

[Bảng SinhVienLamBai (Sinh Viên Làm Bài) 10](#_Toc166859510)

[Bảng ketquahoctap (Kết Quả Học Tập) 10](#_Toc166859511)

[Bảng CotDiem (Cột Điểm) 11](#_Toc166859512)

[Bảng SinhVienKhoaHoc (Sinh Viên Khóa Học) 11](#_Toc166859513)

[2.6. Use-case: 11](#_Toc166859514)

[Mô tả chi tiết Use Case: 11](#_Toc166859515)

[Đăng nhập và Tạo Tài khoản: 11](#_Toc166859516)

[Quản lý hệ thống (Admin): 12](#_Toc166859517)

[Giảng viên: 12](#_Toc166859518)

[Sinh viên: 12](#_Toc166859519)

[CHƯƠNG 3: ĐỀ TÀI QUẢN LÝ HỌC TẬP SINH VIÊN LMS 12](#_Toc166859520)

[1. Các bước thực hiện đề tài 12](#_Toc166859521)

[2. Các chức năng của đề tài: 12](#_Toc166859522)

[Quản trị viên: 12](#_Toc166859523)

[Giảng viên: 12](#_Toc166859524)

[Sinh viên : 13](#_Toc166859525)

[3. Mô tả dữ liệu 13](#_Toc166859526)

[Cơ sở dữ liệu quan hệ: 13](#_Toc166859527)

[4. Mô tả sản phẩm 14](#_Toc166859528)

[KẾT LUẬN 20](#_Toc166859529)

[1. Kết quả đạt được 20](#_Toc166859530)

[2. Hạn chế của đề tài 20](#_Toc166859531)

[3. Phát triển đề tài trong tương lai 20](#_Toc166859532)

[TÀI LIỆU THAM KHẢO 21](#_Toc166859533)

# CHƯƠNG 1: TỔNG QUAN

## 1.Giới thiệu:

### 1.1 Giới thiệu về đề tài

Ngày nay, sự phát triển của công nghệ thông tin đã đem lại những thay đổi lớn trong mọi lĩnh vực, bao gồm cả giáo dục. Việc quản lý thông tin sinh viên và hoạt động giảng dạy đã trở nên phức tạp hơn bao giờ hết, đòi hỏi một hệ thống quản lý hiệu quả và tiện lợi. Để đáp ứng nhu cầu này, đề tài "Xây dựng hệ thống quản lý thông tin sinh viên và giảng dạy" được lựa chọn nhằm cung cấp một giải pháp toàn diện cho các trường đại học và cao đẳng.

Hệ thống sẽ giúp sinh viên và giảng viên thực hiện các hoạt động liên quan đến học tập và giảng dạy một cách dễ dàng hơn. Sinh viên có thể đăng ký khóa học, xem điểm, làm bài tập trực tuyến, trong khi giảng viên có thể tạo bài kiểm tra, nhập điểm và quản lý thông tin về các khóa học. Quản trị viên cũng có thể quản lý toàn bộ hệ thống, từ khóa học, học kỳ đến tài khoản người dùng.

### 1.2 Giới thiệu 1 số chức năng:

**Đăng nhập và Tạo Tài Khoản:**

* Sinh viên và giảng viên có khả năng đăng nhập vào hệ thống.
* Sinh viên có thể tạo tài khoản cá nhân để truy cập các dịch vụ.

**Tính Năng Cho Sinh Viên:**

* Xem điểm: SV có thể xem điểm của mình từ các khóa học đã đăng ký.
* Tra cứu khóa học: SV có thể tìm kiếm thông tin về các khóa học được cung cấp.
* Đăng ký khóa học: SV có thể đăng ký cho các khóa học mình muốn tham gia.
* Làm bài tập: SV có thể thực hiện các bài tập được giao cho từng môn học.

**Tính Năng Cho Giảng Viên:**

* Tạo bài kiểm tra: GV có thể tạo các bài kiểm tra, bài tập cho sinh viên.
* Nhập điểm: Sau khi sinh viên hoàn thành bài kiểm tra,GV có thể nhập điểm vào hệ thống.

**Tính Năng Cho Quản Trị Viên:**

* Quản lý khóa học: QTV có thể thêm, sửa, xóa thông tin về các khóa học.
* Quản lý học kỳ: QTV có thể quản lý các học kỳ, thiết lập thời gian học.
* Quản lý môn học: QTV có khả năng thêm, sửa, xóa thông tin về các môn học.
* Quản lý tài khoản hệ thống: QTV có thể quản lý tài khoản của sinh viên, giảng viên và các thành viên khác trong hệ thống.

## 2. Lý do chọn đề tài:

Đề tài được chọn nhằm giải quyết nhu cầu thực tế trong quản lý giáo dục và phục vụ cho sự tiện lợi trong việc quản lý thông tin sinh viên và hoạt động giảng dạy.

## 3. Yêu cầu đề tài:

Đề tài cần có chức năng làm bài tập tự luận và trắc nghiệm.

Giáo viên phụ trách có thể soạn bài tập và câu hỏi cho sinh viên thực hiện, qua đó tăng tính tương tác và hiệu quả trong quá trình học tập.

# CHƯƠNG 2: CÁC CÔNG NGHỆ ĐƯỢC SỬ DỤNG.

## 2.1. Các tool được sử dụng:

* **Visual Studio Code 2022**

Là một môi trường phát triển tích hợp (IDE) mã nguồn mở, mạnh mẽ và dễ sử dụng do Microsoft phát triển. Nó hỗ trợ nhiều ngôn ngữ lập trình và cung cấp các tính năng như kiểm tra cú pháp, gợi ý mã, và tích hợp git, giúp các lập trình viên tăng năng suất và hiệu quả trong quá trình phát triển phần mềm.

* **Visual Code**
* **Microsoft SQL Server 2019**

Microsoft SQL Server 2019 là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ của Microsoft. Nó cung cấp khả năng lưu trữ, truy vấn và quản lý dữ liệu một cách hiệu quả. SQL Server 2019 hỗ trợ nhiều tính năng tiên tiến như bảo mật cao, tích hợp dữ liệu, và phân tích dữ liệu thời gian thực, giúp quản trị viên dễ dàng quản lý và khai thác thông tin.

* **Postman**

Postman là một công cụ phát triển API phổ biến, cho phép lập trình viên tạo, kiểm thử và quản lý API một cách hiệu quả. Với Postman, người dùng có thể gửi các yêu cầu HTTP, kiểm tra phản hồi, và tự động hóa các kịch bản kiểm thử, giúp đảm bảo tính chính xác và hiệu quả của các API trong quá trình phát triển phần mềm.

## 2.2. Các công nghệ được sử dụng

* **Entity Framework**

Entity Framework (EF) là một ORM (Object-Relational Mapping) được phát triển bởi Microsoft, giúp lập trình viên làm việc với cơ sở dữ liệu bằng cách sử dụng các đối tượng .NET. EF giúp giảm bớt công việc viết các câu lệnh SQL thủ công bằng cách tự động hóa việc ánh xạ giữa các đối tượng và bảng trong cơ sở dữ liệu, nâng cao năng suất và tính dễ bảo trì của ứng dụng.

* **ASP.Net MVC, Web API**

ASP.Net MVC là một khung công tác (framework) của Microsoft dùng để xây dựng các ứng dụng web theo mô hình MVC (Model-View-Controller). Nó giúp phân tách rõ ràng giữa logic nghiệp vụ, giao diện người dùng và điều khiển, làm cho ứng dụng dễ bảo trì và mở rộng. ASP.Net Web API là một phần mở rộng của ASP.Net, dùng để xây dựng các dịch vụ web RESTful, cho phép ứng dụng tương tác với các client khác nhau (web, mobile, v.v.) thông qua các API HTTP.

* **ASP.Net Identity**

ASP.NET Identity là một hệ thống quản lý danh tính linh hoạt và có thể mở rộng, giúp xác thực và quản lý người dùng trong các ứng dụng ASP.NET. ASP.NET Identity hỗ trợ các tính năng như đăng ký, đăng nhập, vai trò và quyền, và tích hợp tốt với các công nghệ bảo mật hiện đại. Hệ thống sẽ sử dụng ASP.NET Identity để xác thực người dùng và cấp cookie, đảm bảo tính bảo mật và tiện lợi trong quản lý người dùng.

* **LINQ(Language Integrated Query**

LINQ là một thành phần của .NET, cho phép thực hiện các truy vấn trên các tập hợp dữ liệu theo cú pháp ngôn ngữ lập trình .NET. LINQ giúp lập trình viên viết các truy vấn dữ liệu ngắn gọn, dễ hiểu và mạnh mẽ, áp dụng trên các nguồn dữ liệu khác nhau như SQL, XML, và các tập hợp trong bộ nhớ. Sử dụng LINQ trong hệ thống này sẽ giúp thực hiện các truy vấn dữ liệu một cách hiệu quả và dễ dàng.một phần mở rộng của ASP.Net, dùng để xây dựng các dịch vụ web RESTful, cho phép ứng dụng tương tác với các client khác nhau (web, mobile, v.v.) thông qua các API HTTP.

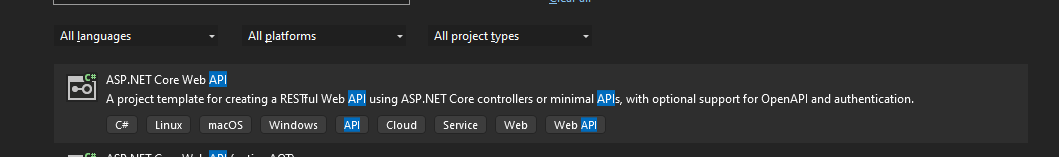
* **Reactjs**

Reactjs là một thư viện JavaScript mã nguồn mở do Facebook phát triển, dùng để xây dựng giao diện người dùng. Reactjs cho phép phát triển các thành phần giao diện độc lập và có thể tái sử dụng, giúp tăng hiệu suất và tính nhất quán của ứng dụng. Với cơ chế Virtual DOM, Reactjs tối ưu hóa việc cập nhật giao diện, giúp ứng dụng phản hồi nhanh và mượt mà.

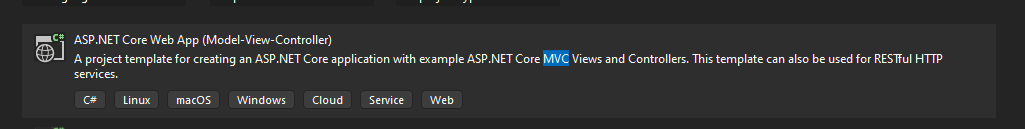
* .**Net 8**

.Net 8 là phiên bản mới nhất của khung công tác .NET do Microsoft phát triển, cung cấp một môi trường nhất quán cho việc phát triển các ứng dụng trên nhiều nền tảng (Windows, macOS, Linux). .Net 8 mang đến nhiều cải tiến về hiệu suất, bảo mật, và tích hợp, hỗ trợ lập trình viên xây dựng các ứng dụng hiện đại, hiệu quả và bảo mật.

## 2.3. Cài đặt:



*Hình 1.1: Cài đặt ASP.NET trên VS*



*Hình 1.2: Cài đặt ASP.NET Core Web App trên VS*

## 2.4. Cấu trúc của hệ thống sử dụng mô hình 3 lớp, gồm DAL, BLL, DTO

**DAL (Data Access Layer - Lớp Truy Cập Dữ Liệu):**

* Lớp này chịu trách nhiệm truy cập vào cơ sở dữ liệu.
* Cung cấp các phương thức để thực hiện các thao tác cơ bản như thêm, sửa, xóa và truy vấn dữ liệu từ cơ sở dữ liệu.
* Đảm bảo tính nhất quán và bảo mật của dữ liệu.

**BLL (Business Logic Layer - Lớp Logic Nghiệp Vụ):**

* Lớp này chứa các logic nghiệp vụ của ứng dụng.
* Xử lý các yêu cầu từ giao diện người dùng và tương tác với lớp DAL để truy xuất dữ liệu từ cơ sở dữ liệu.
* Kiểm tra tính hợp lệ của dữ liệu trước khi thực hiện các thao tác trên cơ sở dữ liệu.
* Điều phối các tác vụ và quyết định luồng dữ liệu trong ứng dụng.

**DTO (Data Transfer Object - Đối Tượng Truyền Dữ Liệu):**

* Lớp này chứa các đối tượng dùng để truyền dữ liệu giữa các lớp trong hệ thống.
* Được sử dụng để đóng gói dữ liệu để truyền từ lớp BLL đến lớp giao diện người dùng hoặc từ lớp DAL đến BLL và ngược lại.
* Dữ liệu được đóng gói vào các đối tượng DTO để giảm thiểu việc truyền thông dữ liệu giữa các lớp và tăng tính module hóa của hệ thống.

**Ưu điểm:**

* **Tường Minh và Tách Biệt:** Phân chia thành từng lớp giúp code trở nên rõ ràng hơn, mỗi lớp đảm nhận một chức năng cụ thể như giao diện, xử lý logic và truy xuất dữ liệu, giảm sự kết dính giữa các phần của hệ thống.
* **Dễ Bảo Trì:** Việc phân chia rõ ràng giúp các thành phần của hệ thống dễ dàng thay đổi mà không ảnh hưởng đến các phần khác, từ việc cô lập thay đổi trong một lớp đến ảnh hưởng tới lớp gần nhất mà không cần sửa đổi toàn bộ chương trình.
* **Dễ Phát Triển và Tái Sử Dụng:** Mô hình 3 lớp tạo ra một chuẩn mực cho việc phát triển, giúp việc thêm chức năng mới trở nên dễ dàng hơn và sử dụng lại mã nguồn được tối ưu hóa, đặc biệt khi chuyển đổi giữa các môi trường như từ Winform sang Webform.
* **Dễ Bàn Giao và Tương Tác:** Sử dụng một quy chuẩn định sẵn giúp việc bàn giao và tương tác giữa các thành viên trong dự án dễ dàng hơn, tiết kiệm thời gian và giảm thiểu rủi ro phát sinh do không hiểu lầm về cách hoạt động của hệ thống.
* **Phân Phối Công Việc:** Việc phân chia rõ ràng trong mô hình 3 lớp giúp kiểm soát và phân phối khối lượng công việc một cách hiệu quả, mỗi nhóm hoặc bộ phận đảm nhận một nhiệm vụ cụ thể, giúp tăng cường hiệu suất làm việc.

### 1. Presentation Layer (GUI):

Có hai thành phần chính sau đây với những tác vụ cụ thể :

UI Components : gồm các thành phần tạo nên giao diện của ứng dụng (GUI). Chúng chịu trách nhiệm thu nhận và hiển thị dữ liệu cho người dùng… Ví dụ : textbox, button, combobox, …

UI Process Components : là thành phần chịu trách nhiệm quản lý các quá trình chuyển đổi giữa các UI… Ví dụ : Sắp xếp quá trình kiểm tra thông tin khách hàng:

1.Hiển thị màn hình tra cứu ID

2.Hiển thị màn hình thông tin chi tiết khách hàng tương ứng

3.Hiển thị màn hình liên lạc với khách hàng.

### 2. Bussiness Layer (BLL) :

Lớp này gồm 4 thành phần:

Service Interface : là thành phần giao diện lập trình mà lớp này cung cấp cho lớp Presentation sử dụng.

Bussiness Workflows : chịu trách nhiệm xác định và điều phối các quy trình nghiệp vụ gồm nhiều bước và kéo dài. Những quy trình này phải được sắp xếp và thực hiện theo một thứ tự chính xác.

Ví dụ : Thực hiện mua một đơn hàng trên tiki qua nhiều bước : kiểm tra gói hàng còn không?, tính tổng chi phí, cho phép giao dịch và sắp xếp việc giao hàng.

Bussiness Components : chịu trách nhiệm kiểm tra các quy tắc nghiệp vụ, ràng buộc logic và thực hiện các công việc . Các thành phần này cũng thực hiện các dịch vụ mà Service Interface cung cấp và Business Workflows sẽ sử dụng nó.

Ví dụ : Tiếp tục ví dụ ở trên. Bạn sẽ cần một Bussiness Component để kiểm tra gói hàng có khả dụng không ? hay một component để tính tổng chi phí,…

Bussiness Entities : thường được sử dụng như Data Transfer Objects ( DTO ) . Bạn có thể sử dụng để truyền dữ liệu giữa các lớp (Presentation và Data Layer). Chúng thường là cấu trúc dữ liệu ( DataSets, XML,… ) hay các lớp đối tượng đã được tùy chỉnh.

Ví dụ : tạo 1 class Student lưu trữ các dữ liệu về tên, ngày sinh, ID, lớp.

### 3. Data Layer (DAL) :

Data Access Logic Components : chịu trách nhiệm chính lưu trữ và truy xuất dữ liệu từ các nguồn dữ liệu (Data Sources) như XML, file system,… Hơn nữa còn tạo thuận lợi cho việc dễ cấu hình và bảo trì.

Service Agents: giúp bạn gọi và tương tác với các dịch vụ từ bên ngoài một cách dễ dàng và đơn giản.

## 2.5. Database:

**A computer screen shot of a computer flow chart

Description automatically generated**

### Tổng Quan

Tài liệu này cung cấp cái nhìn tổng quan chi tiết về sơ đồ cơ sở dữ liệu, bao gồm nhiều bảng liên kết với nhau để quản lý người dùng, khóa học, bài tập, câu hỏi và các mối quan hệ tương ứng của chúng. Trọng tâm chính là quản lý giáo dục, bao gồm vai trò người dùng, phân công khóa học và theo dõi thành tích học tập.

### Các Bảng và Mối Quan Hệ

#### Bảng Users (Người Dùng)

* idUser (INT): Khóa chính.
* username (NVARCHAR(255)): Tên đăng nhập.
* password (NVARCHAR(255)): Mật khẩu.
* firstName (NVARCHAR(255)): Tên.
* lastName (NVARCHAR(255)): Họ.
* email (NVARCHAR(255)): Email.
* role (INT): Vai trò.
* address (NVARCHAR(255)): Địa chỉ.
* avatar (NVARCHAR(255)): Hình đại diện.

**Quan hệ:**

User.idUser < KhoaHoc.idGiangVien: Một người dùng có thể là giảng viên của nhiều khóa học.

User.idUser < SinhVienKhoaHoc.idSinhVien: Một người dùng có thể là sinh viên trong nhiều khóa học.

#### Bảng KhoaHoc (Khóa Học)

* idkhoahoc (INT): Khóa chính.
* idmonhoc (NVARCHAR(255)): Mã môn học.
* idGiangVien (NVARCHAR(255)): Mã giảng viên.
* idHocKy (NVARCHAR(255)): Mã học kỳ.
* tenkhoahoc (NVARCHAR(255)): Tên khóa học.

**Quan hệ:**

KhoaHoc.idHocKy < HocKy.idHocKy: Khóa học thuộc một học kỳ.

KhoaHoc.idmonhoc < monhoc.idmonhoc: Khóa học thuộc một môn học.

KhoaHoc.idGiangVien < User.idUser: Một giảng viên có thể giảng dạy nhiều khóa học.

#### Bảng HocKy (Học Kỳ)

* idHocKy (INT): Khóa chính.
* namHocKy (NVARCHAR(255)): Năm học kỳ.

#### Bảng monhoc (Môn Học)

* idmonhoc (INT): Khóa chính.
* tenmonhoc (NVARCHAR(255)): Tên môn học.
* soTinChi (INT): Số tín chỉ.

#### Bảng baitap (Bài Tập)

* idkhoahoc (INT): Mã khóa học.
* idbaitap (INT): Khóa chính.
* tenbaitap (NVARCHAR(255)): Tên bài tập.
* thoigian (DATE): Thời gian.
* idloaibaitap (INT): Mã loại bài tập.

**Quan hệ:**

baitap.idloaibaitap < loaibaitap.idloaibaitap: Một bài tập thuộc một loại bài tập cụ thể.

baitap.idkhoahoc < KhoaHoc.idkhoahoc: Một bài tập thuộc một khóa học.

baitap.idbaitap < SinhVienLamBai.idBaiTap: Sinh viên có thể làm nhiều bài tập.

#### Bảng loaibaitap (Loại Bài Tập)

* idloaibaitap (INT): Khóa chính.
* tenloaibaitap (NVARCHAR(255)): Tên loại bài tập.

#### Bảng cauhoi (Câu Hỏi)

* idcauhoi (INT): Khóa chính.
* noidung (NVARCHAR(255)): Nội dung câu hỏi.
* idMonHoc (NVARCHAR(255)): Mã môn học.
* idloaicauhoi (INT): Mã loại câu hỏi.

**Quan hệ:**

cauhoi.idMonHoc < monhoc.idmonhoc: Một câu hỏi thuộc một môn học.

cauhoi.idloaicauhoi < loaicauhoi.idloaicauhoi: Một câu hỏi thuộc một loại câu hỏi cụ thể.

cauhoi.idcauhoi < ListTracNghiem.idCauHoi: Một câu hỏi có thể nằm trong nhiều bài kiểm tra trắc nghiệm.

cauhoi.idcauhoi < dapan.idcauhoi: Một câu hỏi có thể có nhiều đáp án.

#### Bảng loaicauhoi (Loại Câu Hỏi)

* idloaicauhoi (INT): Khóa chính.
* tenloaicauhoi (NVARCHAR(255)): Tên loại câu hỏi.

#### Bảng dapan (Đáp Án)

* idcauhoi (INT): Khóa chính.
* iddapan (INT): Mã đáp án.
* noidung (NVARCHAR(MAX)): Nội dung đáp án.
* ketqua (BIT): Kết quả (đúng/sai).

#### Bảng ListTracNghiem (Danh Sách Trắc Nghiệm)

* idTracNghiem (INT): Khóa chính.
* idBaiTap (NVARCHAR(255)): Mã bài tập.
* idCauHoi (NVARCHAR(255)): Mã câu hỏi.

#### Bảng SinhVienLamBai (Sinh Viên Làm Bài)

* idBaiLam (INT): Khóa chính.
* idSinhVien (INT): Mã sinh viên.
* idBaiTap (INT): Mã bài tập.
* idCotDiem (FLOAT): Mã cột điểm.

**Quan hệ:**

SinhVienLamBai.idSinhVien < User.idUser: Một sinh viên có thể làm nhiều bài tập.

SinhVienLamBai.idCotDiem < CotDiem.idCotDiem: Một bài làm có thể có nhiều cột điểm.

#### Bảng ketquahoctap (Kết Quả Học Tập)

* iddiem (INT): Khóa chính.
* idnguoidung (INT): Mã người dùng.
* idkhoahoc (INT): Mã khóa học.
* diemgk (FLOAT): Điểm giữa kỳ.
* diemck (FLOAT): Điểm cuối kỳ.

#### Bảng CotDiem (Cột Điểm)

* idCotDiem (INT): Khóa chính.
* tenCotDiem (INT): Tên cột điểm.
* Diem (FLOAT): Điểm.

#### Bảng SinhVienKhoaHoc (Sinh Viên Khóa Học)

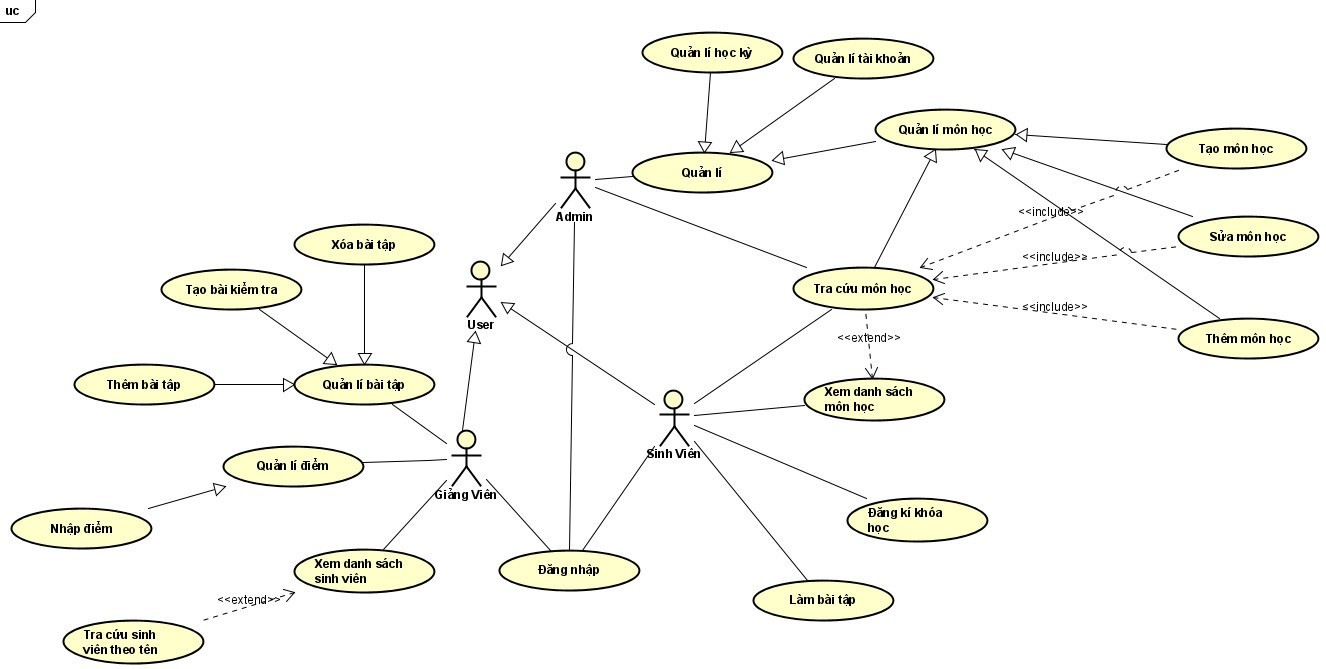
* idSinhVien (INT): Mã sinh viên.
* idSVKhoaHoc (INT): Khóa chính.
* idKhoaHoc (INT): Mã khóa học.

**Quan hệ:**

SinhVienKhoaHoc.idSinhVien < User.idUser: Một sinh viên có thể tham gia nhiều khóa học.

SinhVienKhoaHoc.idKhoaHoc < KhoaHoc.idkhoahoc: Một khóa học có thể có nhiều sinh viên tham gia.

## 2.6. Use-case:

****

### Mô tả chi tiết Use Case:

#### Đăng nhập và Tạo Tài khoản:

* Actors: Quản trị viên, Giảng viên, Sinh viên
* Mô tả: Người dùng có thể đăng nhập vào hệ thống bằng tài khoản của mình. Quản trị viên có thể tạo tài khoản cho giảng viên và sinh viên.

#### Quản lý hệ thống (Admin):

* Quản lý môn học: Thêm, sửa, xóa thông tin về môn học.
* Quản lý khóa học: Thêm, sửa, xóa thông tin về các khóa học.
* Quản lý học kỳ: Quản lý các học kỳ, thiết lập thời gian học.
* Quản lý tài khoản: Quản lý tài khoản người dùng trong hệ thống.

#### Giảng viên:

* Tạo bài kiểm tra: Tạo các bài kiểm tra và bài tập cho sinh viên.
* Nhập điểm: Nhập điểm cho các bài kiểm tra và bài tập của sinh viên.

#### Sinh viên:

* Xem điểm: Xem điểm của mình từ các khóa học đã đăng ký.
* Tra cứu khóa học: Tìm kiếm thông tin về các khóa học được cung cấp.
* Đăng ký khóa học: Đăng ký cho các khóa học mình muốn tham gia.
* Làm bài tập: Thực hiện các bài tập được giao cho từng môn học.

# CHƯƠNG 3: ĐỀ TÀI QUẢN LÝ HỌC TẬP SINH VIÊN LMS

## 1. Các bước thực hiện đề tài

* Bước 1: Phân tích và vẽ sơ đồ usecase.
* Bước 2: Thiết kế và vẽ sơ đồ lớp(UML).
* Bước 3: Chuyển đổi ERD sang mô hình quan hệ.
* Bước 3: Thiết kế lệnh SQL và thực hiện tạo các bảng trong cơ sở dữ liệu.
* Bước 4: Thực hiện tạo dự án với .Net và tạo các lớp DTO tương ứng trong cơ sở dữ liệu và cài đặt cấu hình kết nối.
* Bước 5: Viết lệnh truy vấn tương tác với cơ sở dữ liệu trong tầng DAL và đổ ra cho tầng BLL(tầng dịch vụ).
* Bước 6: Tạo các controller để sử dụng API và tạo trang Swagger.
* Bước 7: Tạo trang Admin(MVC) và sử dụng các dịch vụ BLL để quản lý hệ thống.
* Bước 8: Lập trình giao diện cho hệ thống sử dụng Reactjs.

## 2. Các chức năng của đề tài:

### Quản trị viên:

* + Quản lý môn học, khóa học, học kỳ, tài khoản người dùng các chức năng như thêm xóa sửa.

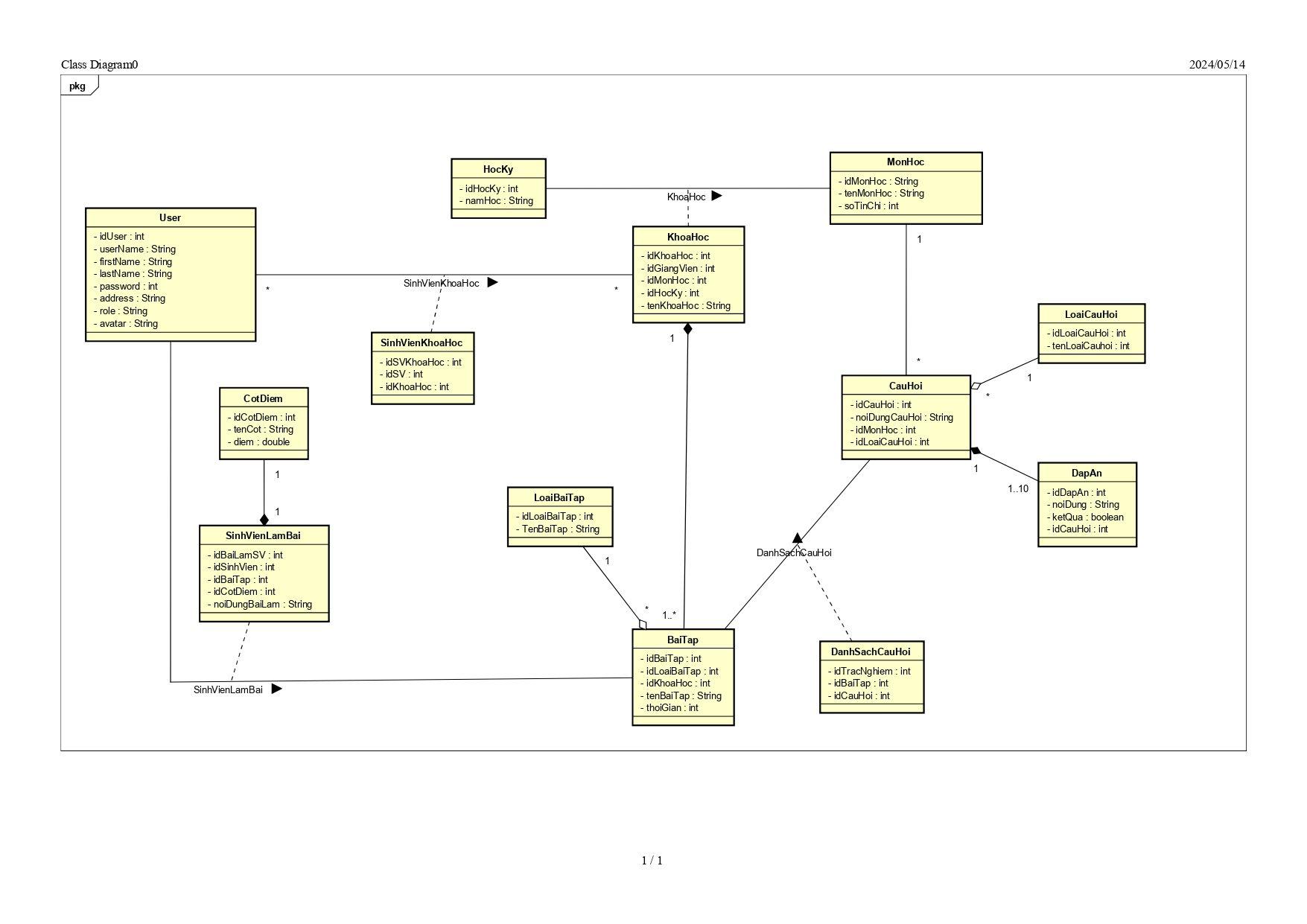
### Giảng viên:

* + Đăng nhập, thêm xóa bài tập, soạn danh sách câu hỏi tự luận/trắc nghiệm, nhập điểm cho sinh viên(bài tự luận), xem danh sách sinh viên tra cứu sinh viên theo tên.

### Sinh viên :

* + Đăng ký/Đăng nhập, đăng ký khóa học/bỏ đăng ký khóa học, làm bài trắc nghiệm, làm bài tự luận, xem thông tin điểm trong khóa học, xuất bảng điểm CSV.

## 3. Mô tả dữ liệu



### Cơ sở dữ liệu quan hệ:

● User(idUser, username, firstname, lastname, password, address, role, avatar)

● MonHoc(idMonHoc, tenMonHoc, soTinChi)

● HocKy(idHocKy, namHoc)

● KhoaHoc(idKhoaHoc, idGiangVien, idMonHoc, idHocKy, tenKhoaHoc)

● SinhVienKhoaHoc(idSVKhoaHoc, idSV, idKhoaHoc)

● LoaiCauHoi(idLoaiCauHoi,tenLoaiCauHoi)

● CauHoi(idCauHoi, noiDungCauHoi,idMonHoc,idLoaiCauHoi)

● DapAn(idDapAn, noiDung, ketQua, idCauHoi)

● LoaiBaiTap(idLoaiBaiTap)

● BaiTap(TenBaiTap)

● DanhSachCauHoi(idTracNghiem,idBaiTap, idCauHoi)

● CotDiem(idCotDiem, tenCot, Diem)

● SinhVienLamBai(idBaiLamSV,idSinhVien,idBaiTap,idCotDiem, noiDungBaiLam)

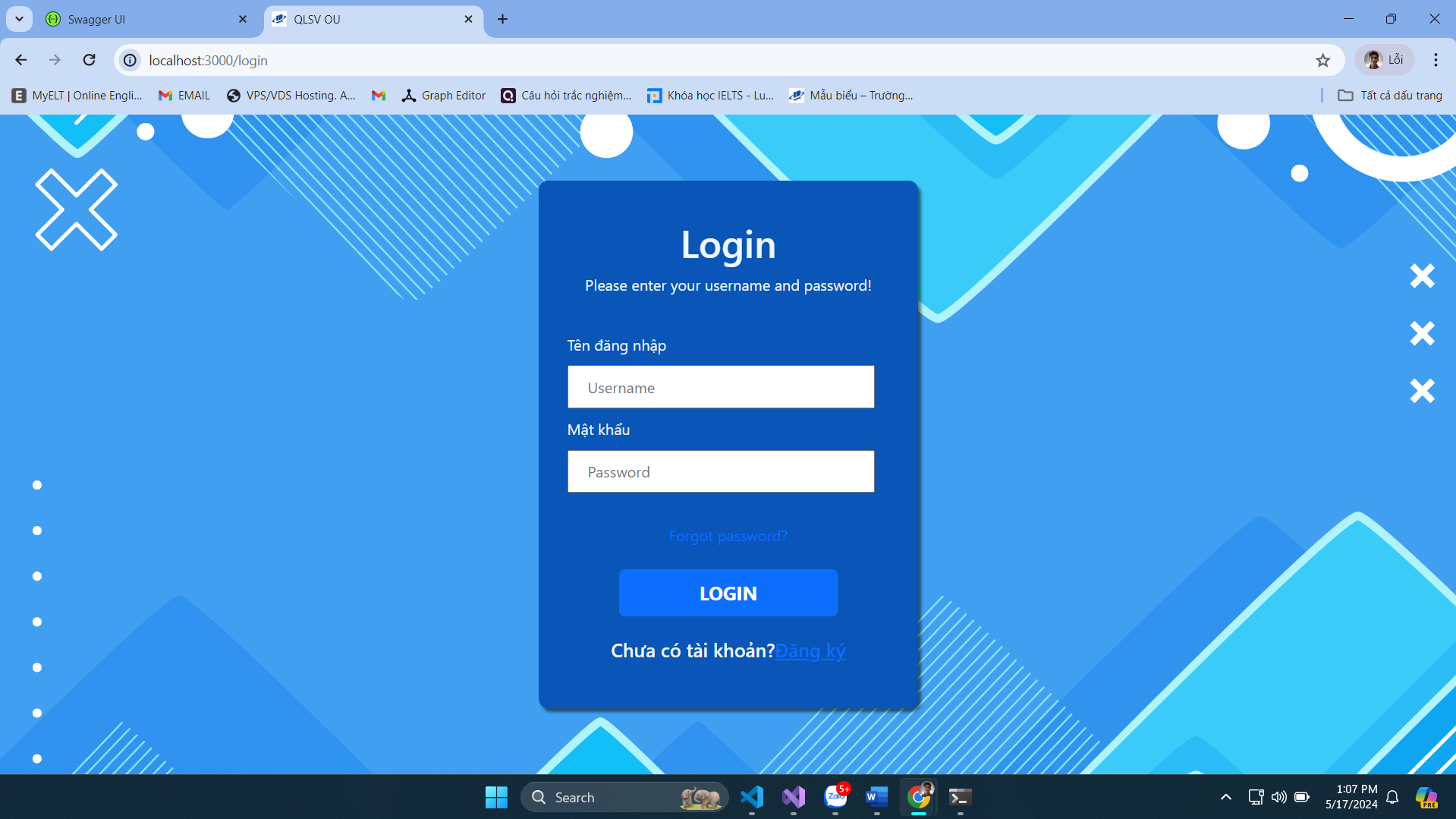
## 4. Mô tả sản phẩm

Hệ thống web quản lý học tập cho sinh viên là một nền tảng trực tuyến được thiết kế để hỗ trợ quá trình học tập và quản lý học vụ. Hệ thống này cung cấp nhiều tính năng đa dạng nhằm giúp sinh viên, giảng viên và quản trị viên tối ưu hóa công việc của mình. Cụ thể, sinh viên có thể truy cập vào khóa học, nộp bài tập, xem điểm số và làm bài tập một cách dễ dàng và nhanh chóng. Ngoài ra, hệ thống cũng cho phép sinh viên tham gia vào các diễn đàn thảo luận.

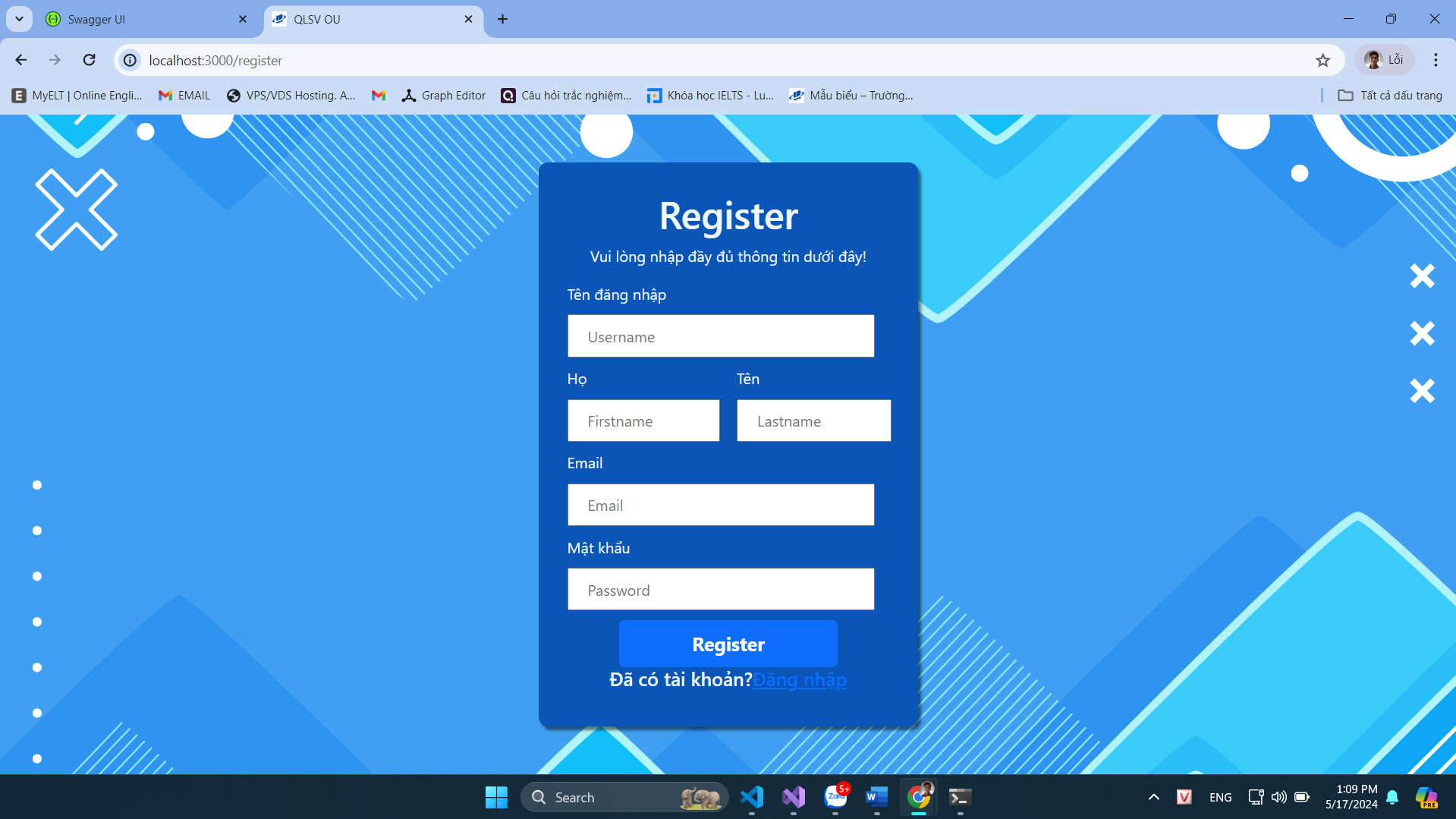
Đối với giảng viên, hệ thống này cung cấp các công cụ để tạo và quản lý các khóa học, tổ chức các bài giảng trực tuyến, soạn câu hỏi và bài tập, chấm điểm và đưa ra nhận xét cho từng bài tập của sinh viên. Giảng viên cũng có thể theo dõi tiến độ học tập của sinh viên và cung cấp hỗ trợ kịp thời khi cần thiết.

Quản trị viên của hệ thống có thể quản lý toàn bộ dữ liệu học tập, theo dõi hoạt động của sinh viên và giảng viên, và đảm bảo hệ thống hoạt động trơn tru. Họ cũng có thể tạo báo cáo và phân tích dữ liệu để cải thiện chất lượng giảng dạy và học tập.

Tóm lại, hệ thống web quản lý học tập cho sinh viên là một công cụ toàn diện và hiệu quả, giúp tối ưu hóa quá trình giáo dục thông qua việc cung cấp một môi trường học tập linh hoạt, tiện lợi và tương tác.

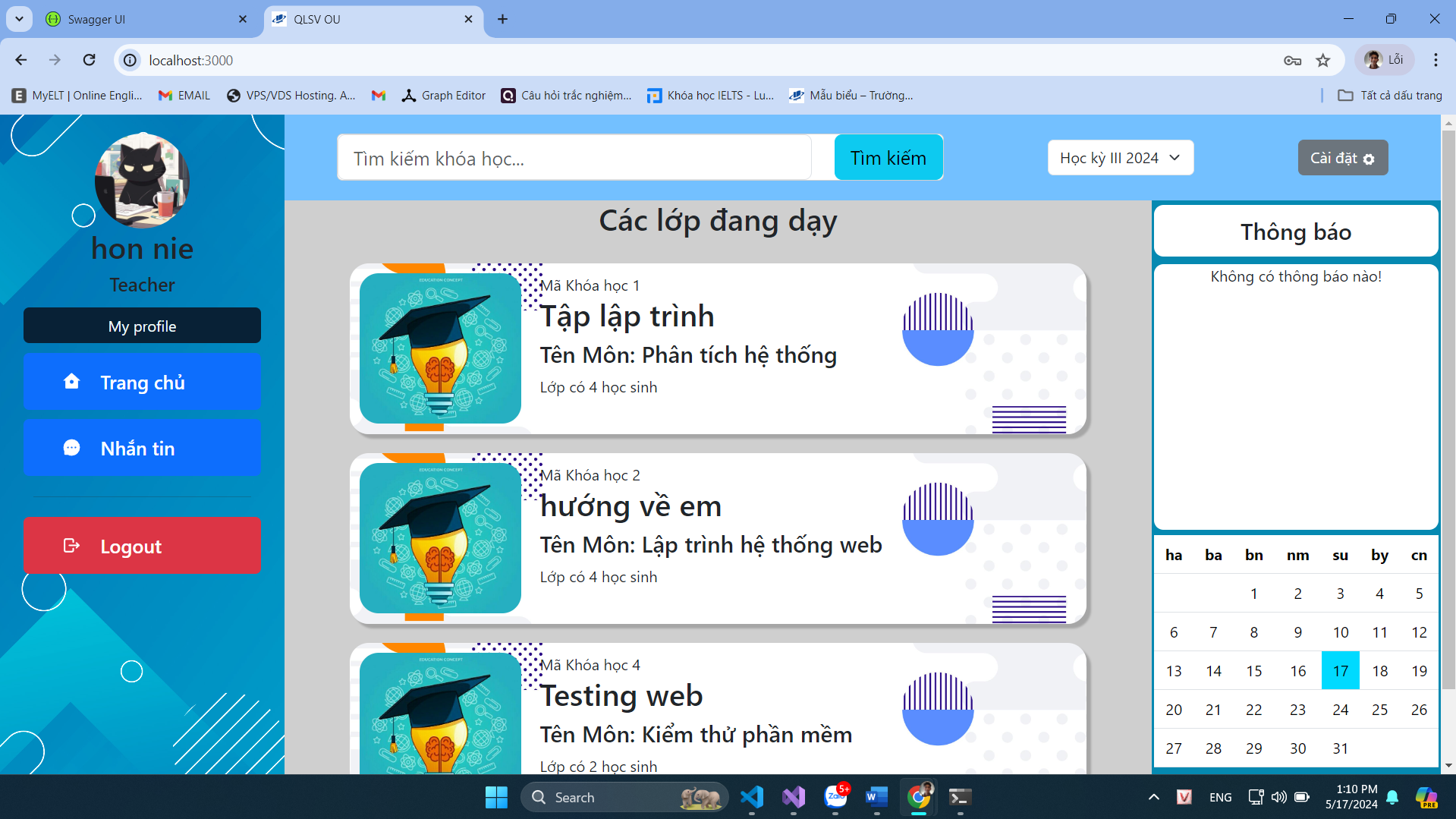


*Hinh 4.1 Giao diện đăng nhập.*

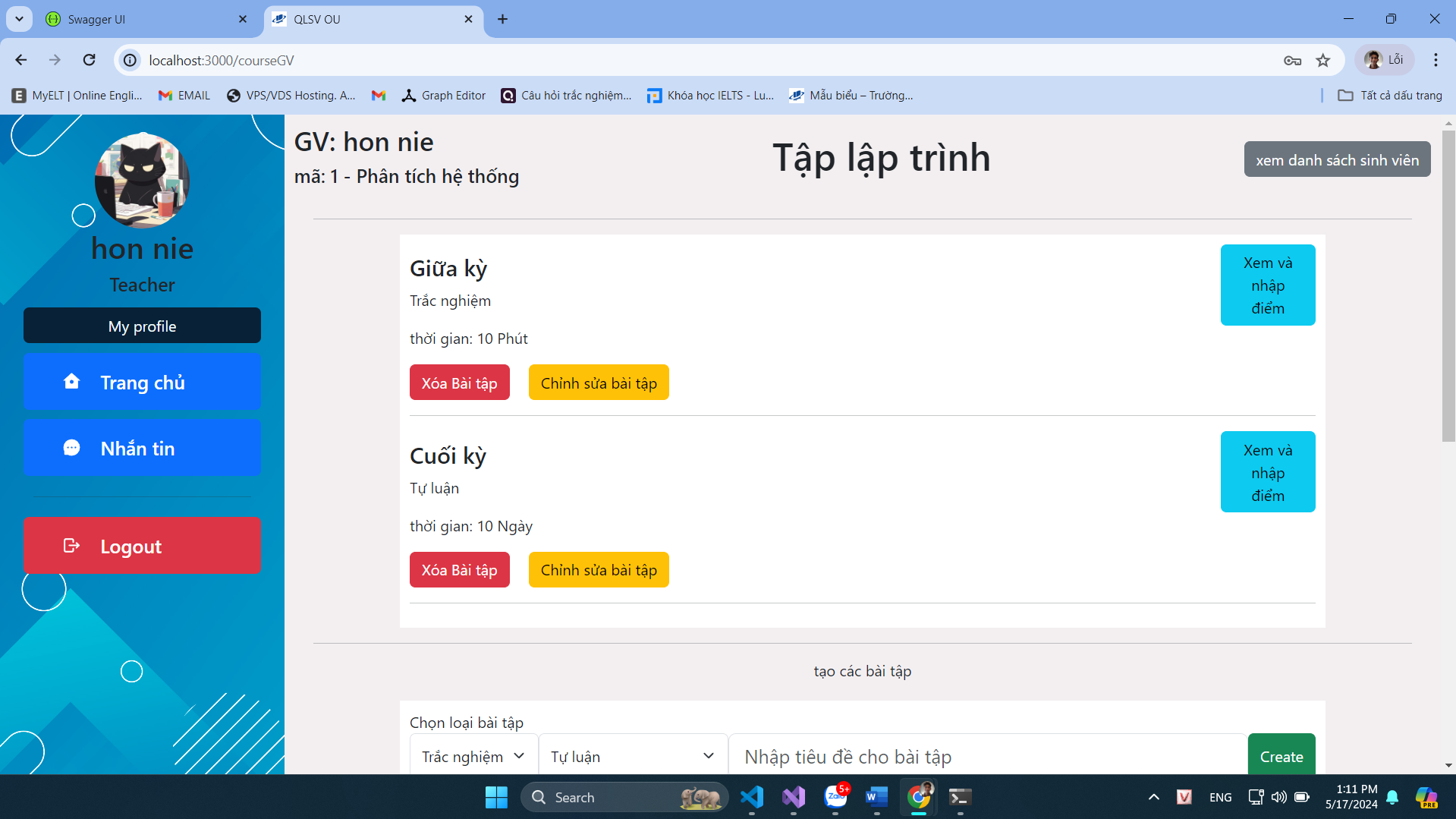
**

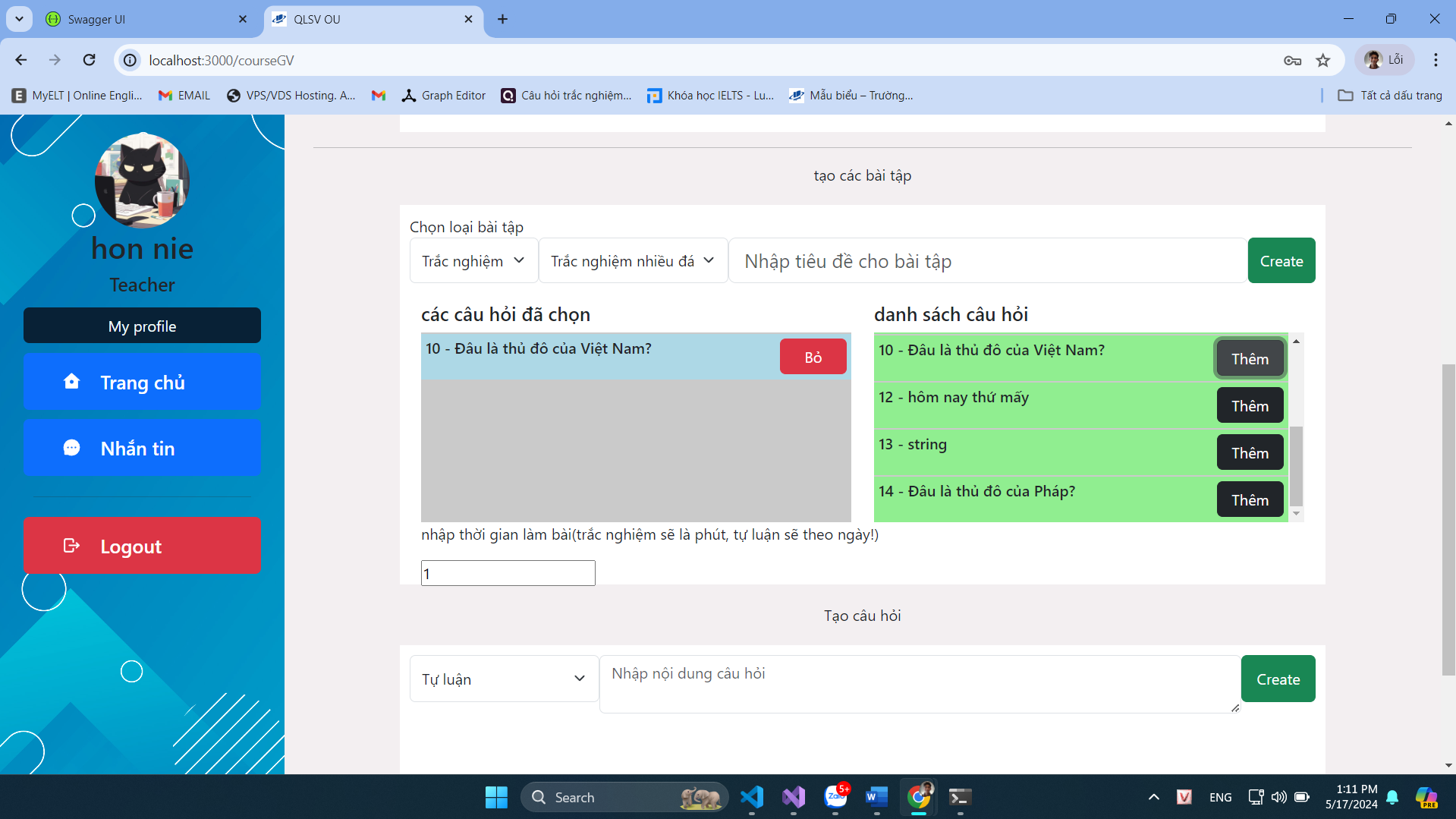
*Hinh 4.2 Giao diện đăng ký.*

**Giảng viên:**

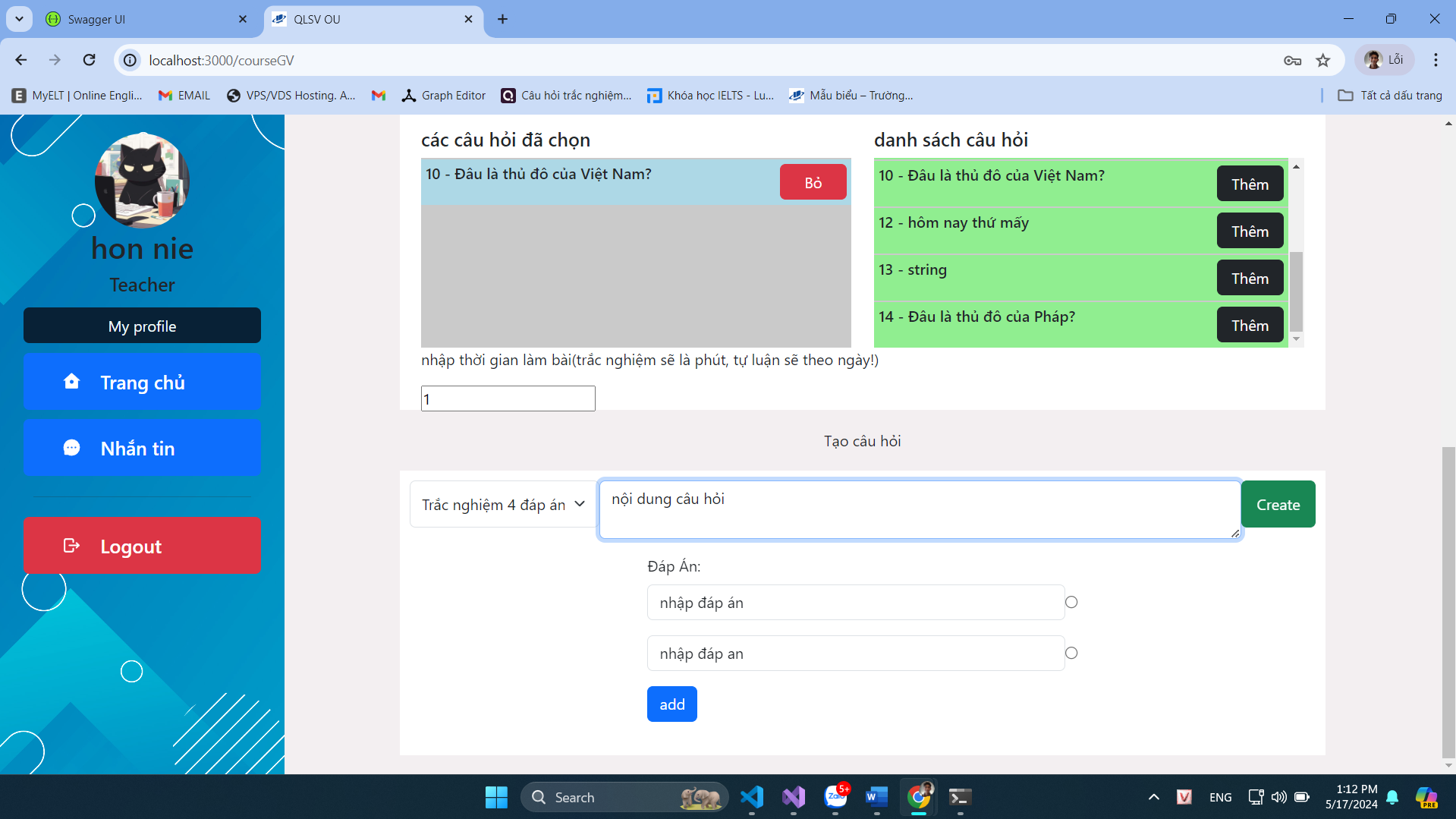
****

*Hinh 4.3 Giao diện trang chủ.*

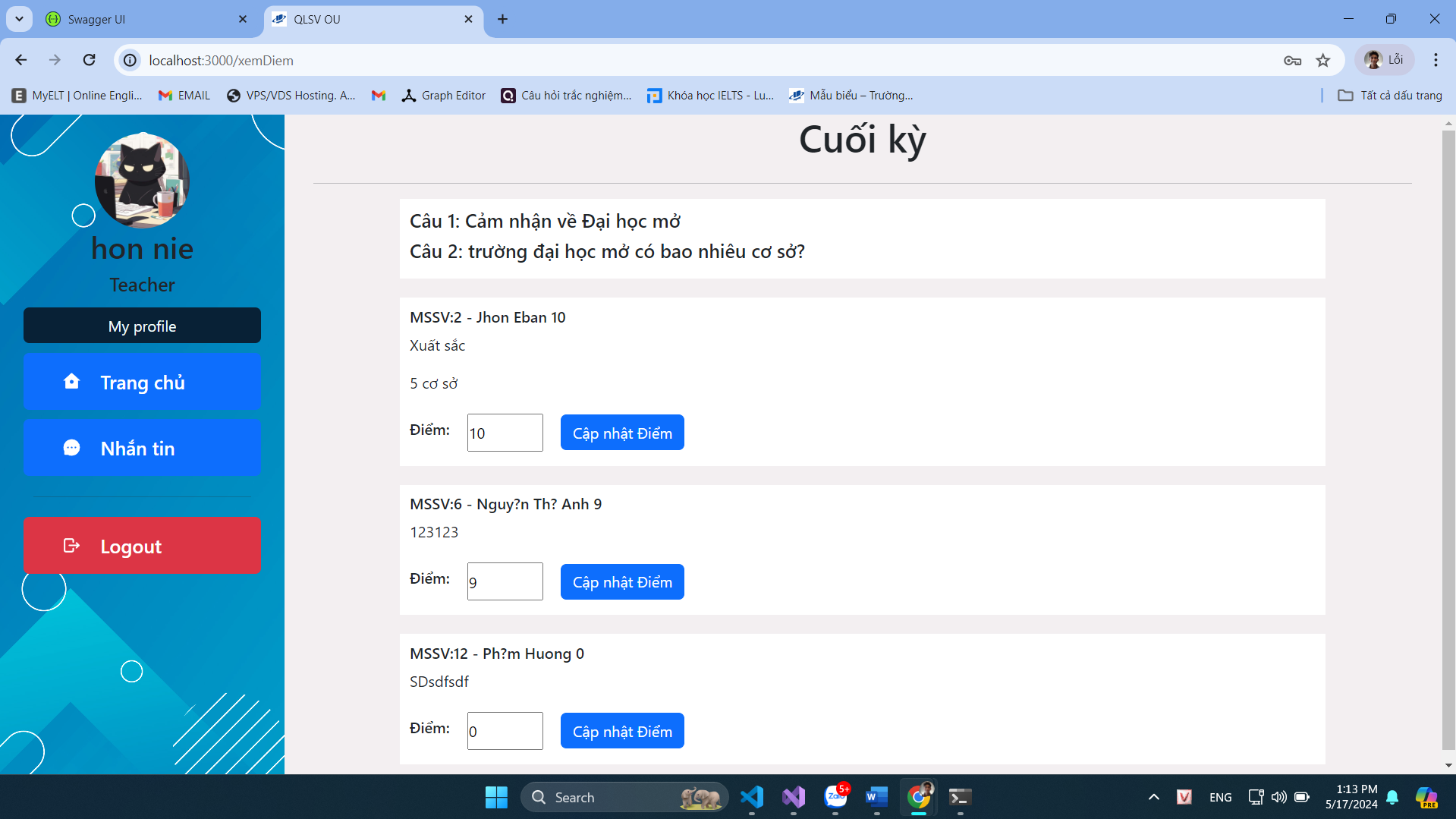
****

*Hinh 4.4 giao diện khóa học.*****

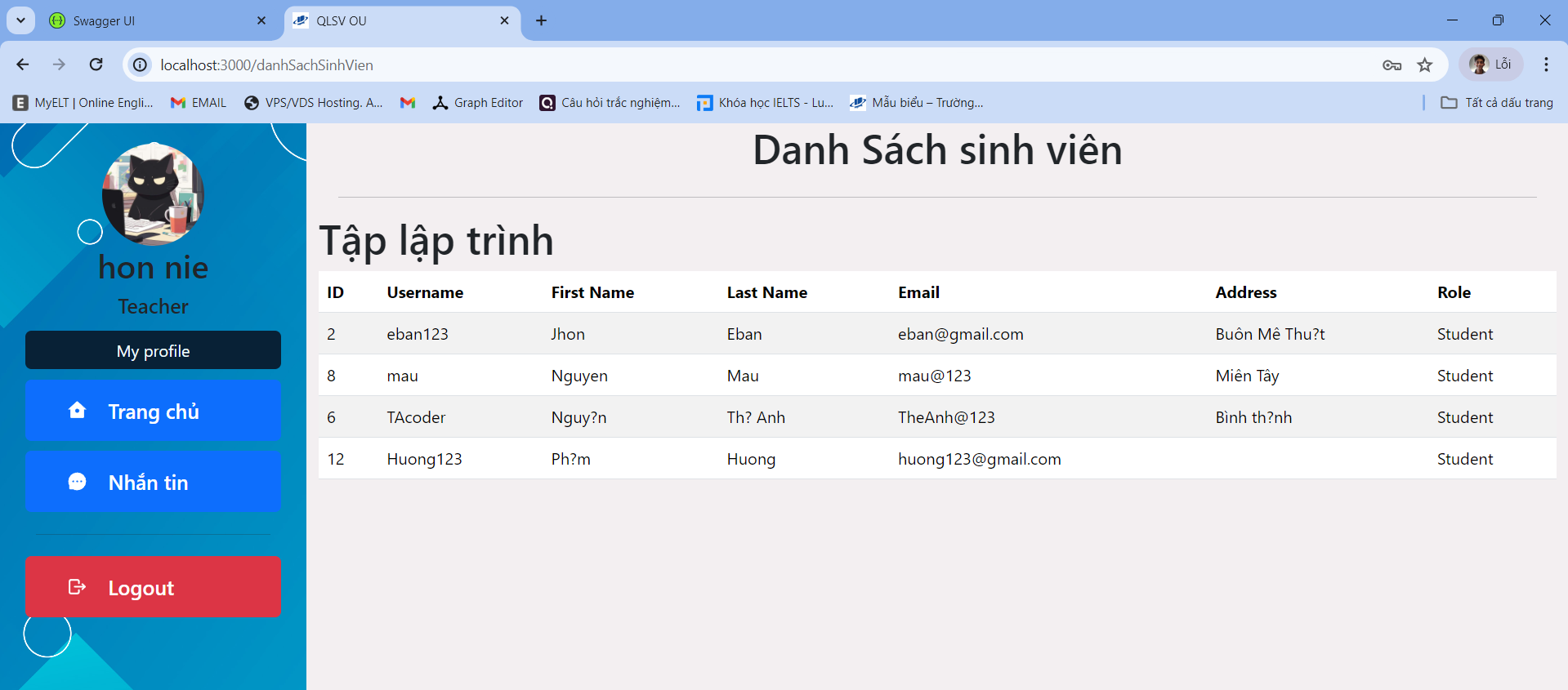
*Hinh 4.5 Giao diện thêm bài tập.*

****

*Hinh 4.6 Giao diện thêm câu hỏi.*

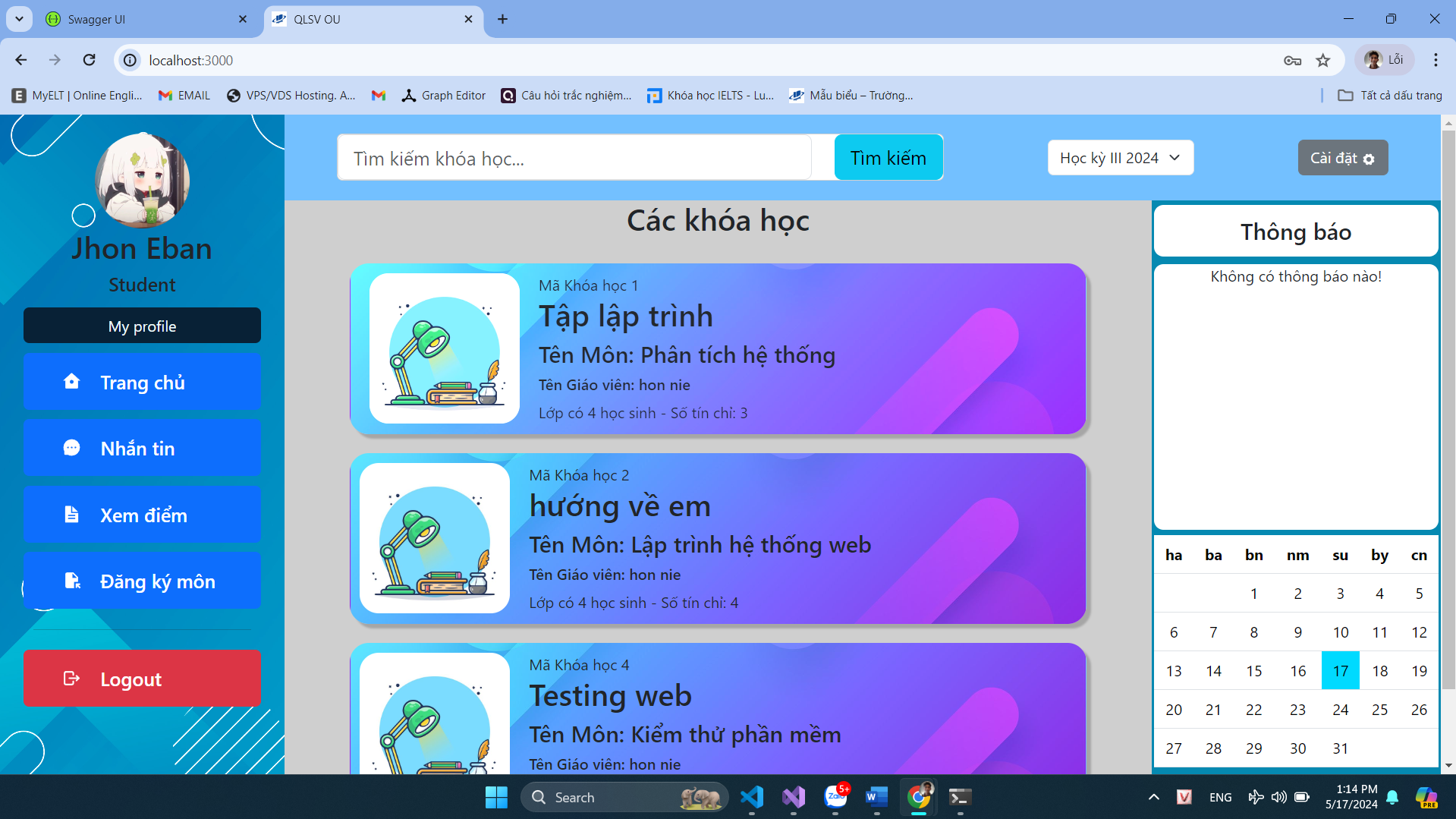
****

*Hinh 4.7 Giao diện nhập điểm trắc nghiệm.*

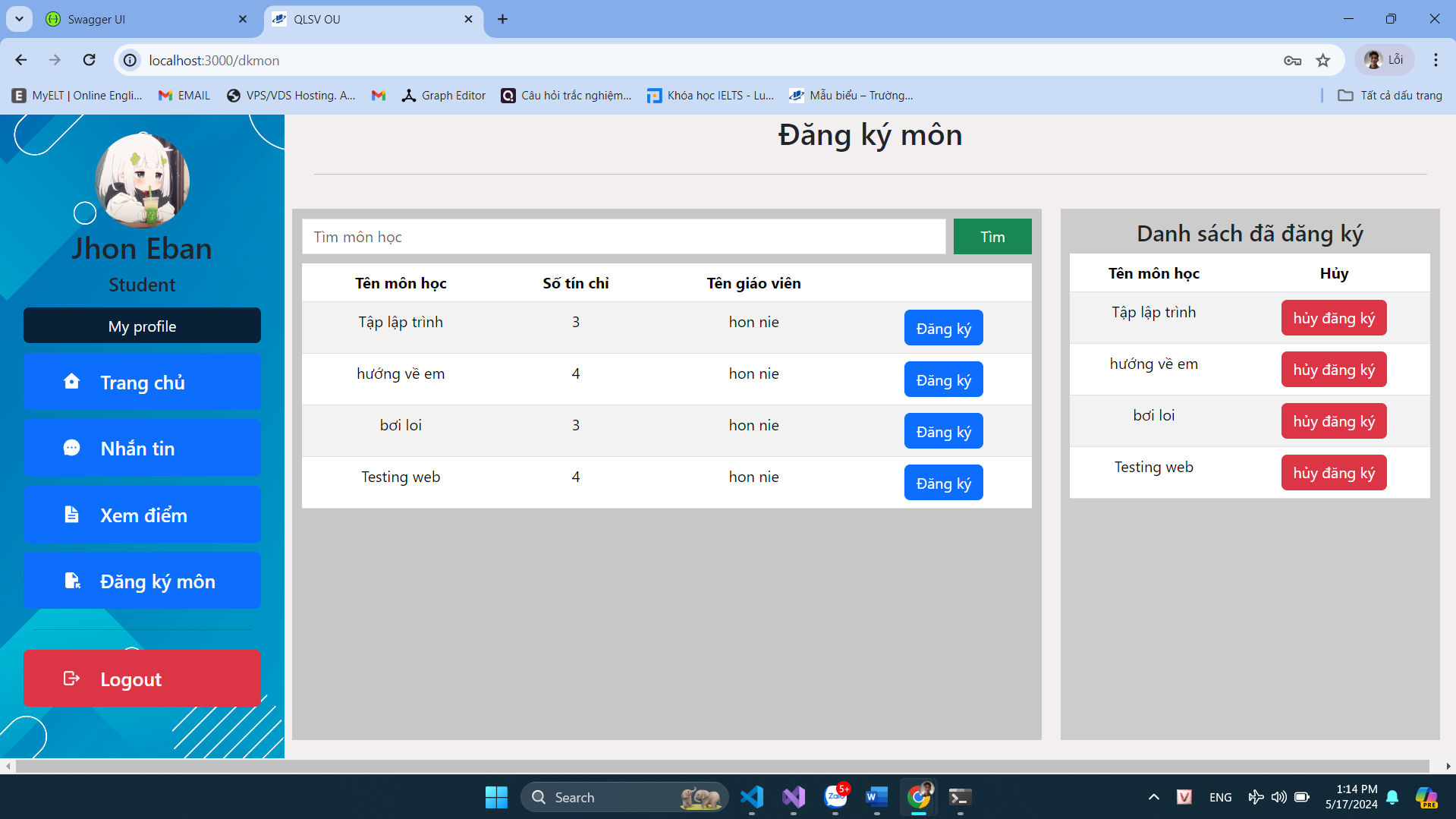
****

*Hinh 4.8 Giao diện xem danh sách sinh viên.*

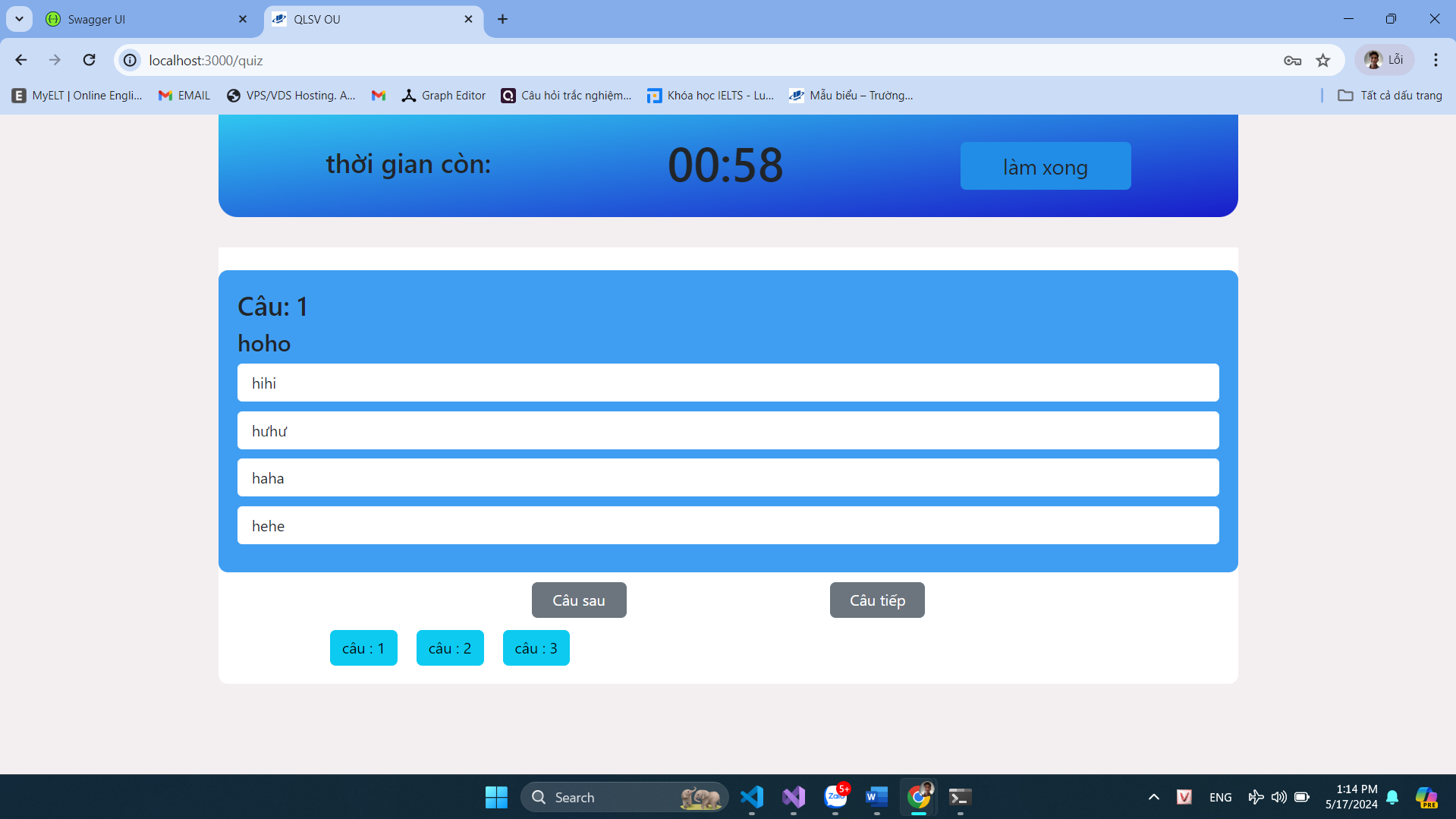
**Sinh viên:**

****

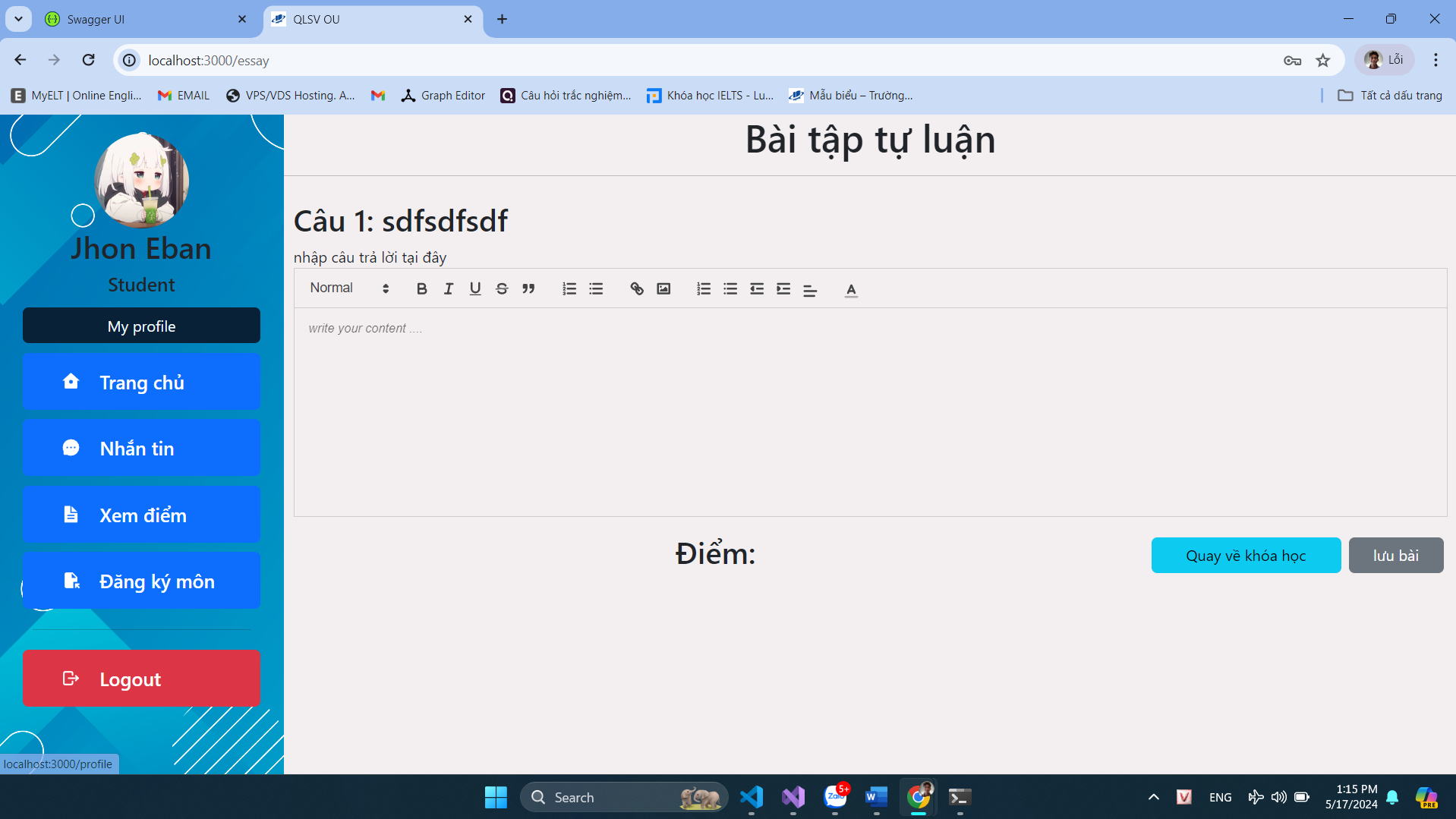
*Hinh 4.9 Giao diện chỉnh sinh viên.*

****

*Hinh 4.10 Giao diện đăng ký khóa học.*

****

*Hinh 4.11 Giao diện làm bài trắc nghiệm.*

****

*Hinh 4.1 Giao diện làm bài tự luận.*

## KẾT LUẬN

### 1. Kết quả đạt được

Hệ thống đã tạo ra một cơ sở dữ liệu linh hoạt và dễ dàng truy cập, giúp sinh viên cập nhật thông tin và thực hiện bài tập một cách thuận tiện.

### 2. Hạn chế của đề tài

**Hạn chế về Thiết Bị**: Hiện tại, hệ thống chỉ có sẵn trên máy tính xách tay, chưa tối ưu hóa cho máy tính bảng và điện thoại di động.

**Tính Bảo Mật Chưa Đảm Bảo:** Mặc dù đã có hệ thống đăng nhập, nhưng cần cải thiện để đảm bảo tính an toàn của thông tin cá nhân và dữ liệu.

### 3. Phát triển đề tài trong tương lai

**Chức Năng Nhắn Tin Trực Tuyến:** Thêm chức năng chat trực tiếp giữa giảng viên và sinh viên giúp dễ dàng hơn trong việc trao đổi thông tin, giải đáp thắc mắc.

**Diễn Đàn Trao Đổi Học Tập:** Tạo một diễn đàn trao đổi giữa sinh viên và giảng viên để hỗ trợ trao đổi kiến thức, thảo luận về nội dung học tập.

**Chức Năng Nhập Điểm CSV:** Thêm tính năng nhập điểm từ file CSV giúp quản lý điểm một cách nhanh chóng và hiệu quả hơn.

## TÀI LIỆU THAM KHẢO

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Công việc** | **Mức độ hoàn thành** |
| Y Jhon ÊBan | Vẽ UML, Thiết kế giao diện, lập trình giao diện, viết API, lập trình giao diện Admin, quản lý API user, khóa học, môn học, học kỳ | 100% |
| Nguyễn Văn Mau | Tạo truy vấn dữ liệu, Quản lý API, cột điểm, câu hỏi, khóa học, | 100% |
| Nguyễn Trần Đức Anh | Thiết kế Database, lập trình giao diện Đăng Ký, Khởi tạo Dữ liệu cho database, vẽ use case | 100% |