BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN TỐT NGHIỆP**

**XÂY DỰNG NÂNG CẤP WEBSITE CHẤM BÀI LẬP TRÌNH THUẬT TOÁN TỰ ĐỘNG NTUCODER**

**(WEBSITE LMS HỌC LẬP TRÌNH)**

|  |  |
| --- | --- |
| **Giáo viên hướng dẫn** | **: ThS. Trần Minh Văn** |
| **Sinh viên thực hiện** | **: Nguyễn Thiết Duy Đan** |
| **Mã số sinh viên** | **: 63133615** |

KHÁNH HÒA - 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN TỐT NGHIỆP**

**XÂY DỰNG NÂNG CẤP WEBSITE CHẤM BÀI LẬP TRÌNH THUẬT TOÁN TỰ ĐỘNG NTUCODER**

**(WEBSITE LMS HỌC LẬP TRÌNH)**

|  |  |
| --- | --- |
| GVHD | : ThS. Trần Minh Văn |
| SVTH | : Nguyễn Thiết Duy Đan |
| MSSV | : 63133615 |

KHÁNH HÒA - 2025

**TRƯỜNG ĐẠI HỌC NHA TRANG**

Khoa/Viện: …………………………….

**PHIẾU ĐĂNG KÝ NHẬN ĐỒ ÁN/KHÓA LUẬN/CHUYÊN ĐỀ TỐT NGHIỆP**

**1. Họ và tên sinh viên/nhóm sinh viên đăng ký đề tài** (sĩ số trong nhóm:……) (1)……………………………………………..MSSV:……………………khóa:…… (2)……………………………………………..MSSV:……………………khóa:…… (3)……………………………………………..MSSV:……………………khóa:…… (4)……………………………………………..MSSV:……………………khóa:…… (5)……………………………………………..MSSV:……………………khóa:…… Ngành:………………………………………...Khoa/Viện:…………………………..

**2. Tên đề tài đăng ký:**

……………………………...……………………………………………………………

……………………………...……………………………………………………………

……………………………...……………………………………………………………

……………………………...……………………………………………………………

**3. Cán bộ hướng dẫn:**

(1) ……………………………………………………………………………………….

(2) ……………………………………………………………………………………….

Sinh viên đã hiểu rõ yêu cầu của đề tài, cam kết thực hiện và hoàn thành theo đúng nội dung, thời hạn đề ra.

*Khánh Hòa, ngày… tháng … năm …….*  
  **Ý kiến cán bộ hướng dẫn Sinh viên**  
 *(Ký và ghi rõ họ tên) (Ký và ghi rõ họ tên)*

**Trưởng Bộ môn duyệt**  
*(Ký và ghi rõ họ tên)*

**TRƯỜNG ĐẠI HỌC NHA TRANG**

Khoa/Viện: …………………………….

**PHIẾU THEO DÕI TIẾN ĐỘ VÀ ĐÁNH GIÁ  
ĐỒ ÁN/KHÓA LUẬN/CHUYÊN ĐỀ TỐT NGHIỆP**  
**(Dùng cho CBHD và nộp cùng báo cáo ĐA/KL/CĐTN của sinh viên)**

Tên đề tài: .........................................................................................................................

Giảng viên hướng dẫn: .....................................................................................................

Sinh viên được hướng dẫn: ……………………………….………MSSV: .....................

Khóa: ………………… Ngành: .......................................................................................

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Lần KT* | *Ngày* | *Nội dung* | *Nhận xét của GVHD* | |
| 1 |  |  |  | |
| 2 |  |  |  | |
| 3 |  |  |  | |
| 4 |  |  |  | |
| 5 |  |  |  | |
| **Kiểm tra giữa tiến độ của Trưởng Bộ môn** | | | | |
| Ngày kiểm tra:  ……………...……… | | Đánh giá công việc hoàn thành: ……%:  Được tiếp tục:  Không tiếp tục:  | | Ký tên  ………………………. |
| *Lần KT* | *Ngày* | *Nội dung* | *Nhận xét của GVHD* | |
| 6 |  |  |  | |
| 7 |  |  |  | |
| 8 |  |  |  | |
| 9 |  |  |  | |
| … |  |  |  | |

**Nhận xét chung** (sau khi sinh viên hoàn thành ĐA/KL/CĐTN):

………………..…………………………………………………………………….……

…………………………………………………………………..………….……………

…………………………………………..………………………………………….……

Điểm hình thức: ……/10 Điểm nội dung: ......./10 **Điểm tổng kết**: ………/10

*Đối với ĐA/KLTN:* Kết luận sinh viên: Được bảo vệ:  Không được bảo vệ: 

*Khánh Hòa, ngày……. tháng…… năm …….*  
**Cán bộ hướng dẫn***(Ký và ghi rõ họ tên)*

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**Khoa/Viện: ……………………………….**

**PHIẾU CHẤM ĐIỂM CHUYÊN ĐỀ TỐT NGHIỆP**

***(Dành cho cán bộ chấm phản biện)***

Tên đề tài:

Chuyên ngành:

Họ và tên sinh viên: …………………………Mã sinh viên: ...................................

Người phản biện (học hàm, học vị, họ và tên):

Cơ quan công tác:

**Phần đánh giá và cho điểm của người phản biện (tính theo thang điểm 10)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tiêu chí**  **đánh giá** | **Trọng số (%)** | **Mô tả mức chất lượng** | | | | **Điểm** |
| **Giỏi** | **Khá** | **Đạt yêu cầu** | **Không đạt** |
| **9 - 10** | **7 - 8** | **5 - 6** | **< 5** |
| Hình thức bản  thuyết minh | 30 |  |  |  |  |  |
| Nội dung bản  thuyết minh | 30 |  |  |  |  |  |
| Kết quả  nguyên cứu | 20 |  |  |  |  |  |
| Mức độ trích dẫn và  sao chép | 20 |  |  |  |  |  |
| **ĐIỂM TỔNG:** | | | | | |  |

*Ghi chú: Điểm tổng làm tròn đến 1 số lẻ.*

*Khánh Hòa, ngày…….tháng………năm………..*

**Cán bộ chấm phản biện**

*(Ký và ghi rõ họ tên)*

LỜI CAM ĐOAN

Em xin cam đoan rằng toàn bộ nội dung trong báo cáo đồ án tốt nghiệp với đề tài: **“Xây dựng nâng cấp website chấm bài lập trình thuật toán tự động NTUCoder (Website LMS học lập trình)”** là kết quả nghiên cứu, tìm hiểu và thực hiện nghiêm túc của cá nhân em dưới sự hướng dẫn tận tình của giảng viên hướng dẫn.

Tất cả thông tin, số liệu, mã nguồn, tài liệu tham khảo được trình bày trong báo cáo đều được thu thập, trích dẫn rõ ràng, minh bạch và tuân thủ quy tắc đạo đức học thuật. Sản phẩm cuối cùng được xây dựng dựa trên kiến thức, kỹ năng cá nhân và không sao chép hay sử dụng trái phép bất kỳ đồ án, luận văn hay sản phẩm nào đã có trước đó. Nếu phát hiện bất kỳ hành vi sao chép, gian lận hoặc vi phạm bản quyền trong nội dung hoặc sản phẩm của đề tài, em xin hoàn toàn chịu trách nhiệm trước nhà trường và pháp luật.

Em xin chân thành cảm ơn sự hỗ trợ và tạo điều kiện từ phía nhà trường, thầy cô giảng viên và những người đã giúp đỡ em trong quá trình thực hiện đề tài.

*Khánh Hòa, ngày…….tháng…….năm………*

**Sinh viên thực thiện**

**Nguyễn Thiết Duy Đan**

LỜI CẢM ƠN

Trong quá trình học tập và thực hiện khóa luận tốt nghiệp, em đã nhận được rất nhiều sự quan tâm, hỗ trợ và chỉ dẫn tận tình từ các thầy cô trong trường. Để hoàn thành được đề tài ***“Xây dựng nâng cấp website chấm bài lập trình thuật toán tự động NTUCoder (website LMS học lập trình)”***, em xin được bày tỏ lòng biết ơn sâu sắc đến quý thầy cô đã luôn đồng hành, tạo điều kiện thuận lợi cho em trong suốt thời gian qua.

Trước hết, em xin gửi lời cảm ơn chân thành đến **Ban giám hiệu trường Đại học Nha Trang**, đã tạo dựng một môi trường học tập năng động, hiện đại, là nền tảng để sinh viên có cơ hội tiếp cận với tri thức mới, ứng dụng thực tế và phát triển kỹ năng chuyên môn. Em cũng xin gửi lời cảm ơn sâu sắc đến **quý thầy cô trong Khoa Công nghệ thông tin**, những người đã tận tâm giảng dạy, truyền đạt cho em những kiến thức quý báu không chỉ về chuyên ngành mà còn về tư duy, phương pháp học tập và làm việc. Nhờ có sự dìu dắt, định hướng từ quý thầy cô trong suốt các học kỳ, em đã có thể tự tin triển khai đề tài đồ án này một cách bài bản và có định hướng rõ ràng hơn.

Đặc biệt, em xin trân trọng cảm ơn **thầy Trần Minh Văn** – giảng viên hướng dẫn của em, người đã luôn nhiệt tình hỗ trợ, đưa ra những góp ý thiết thực trong suốt quá trình làm khóa luận. Thầy không chỉ giúp em định hình được hướng phát triển của đề tài mà còn giúp em nâng cao tính thực tiễn, cải thiện giao diện và xử lý hiệu quả hơn trong sản phẩm. Sự tận tâm và nghiêm túc của thầy chính là nguồn động lực để em cố gắng hoàn thiện đồ án một cách tốt nhất.

Trong quá trình thực hiện, mặc dù em đã cố gắng nỗ lực hết sức, nhưng do kinh nghiệm thực tế và năng lực nghiên cứu còn hạn chế nên không tránh khỏi những sai sót và thiếu sót. Em rất mong nhận được những góp ý, phản hồi từ quý thầy cô để em có thể tiếp tục hoàn thiện bản thân và nâng cao năng lực chuyên môn trong tương lai.

Một lần nữa, em xin kính chúc quý thầy cô thật nhiều sức khỏe, niềm vui và luôn thành công trong sự nghiệp trồng người cao quý.

Em xin chân thành cảm ơn!

MỤC LỤC

[LỜI CAM ĐOAN 1](#_Toc199868681)

[LỜI CẢM ƠN 2](#_Toc199868682)

[MỤC LỤC 3](#_Toc199868683)

[DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT 5](#_Toc199868684)

[DANH MỤC HÌNH ẢNH 6](#_Toc199868685)

[DANH MỤC BẢNG BIỂU 9](#_Toc199868686)

[PHẦN MỞ ĐẦU 11](#_Toc199868687)

[Chương 1. TỔNG QUAN VỀ WEBSITE NTU-CODER-LMS 13](#_Toc199868688)

[1.1 MÔ TẢ BÀI TOÁN 13](#_Toc199868689)

[1.2 XÂY DỰNG PHƯƠNG PHÁP VÀ NỘI DUNG NGHIÊN CỨU 14](#_Toc199868690)

[1.2.1 Phương pháp nghiên cứu 14](#_Toc199868691)

[1.2.2 Nội dung nghiên cứu 14](#_Toc199868692)

[1.2.3 Lý do lựa chọn phương pháp nghiên cứu 15](#_Toc199868693)

[1.3 KHẢO SÁT, ĐÁNH GIÁ THỰC TRẠNG VÀ TRẢI NGHIỆM LIÊN QUAN ĐẾN HỆ THỐNG NGHIÊN CỨU 15](#_Toc199868694)

[1.3.1 Khảo sát trải nghiệm người dùng đối với trang website NTUCoder 16](#_Toc199868695)

[1.3.2 Trải nghiệm cảm giác quản lý trang website NTUCoder 16](#_Toc199868696)

[1.4 CẤU TRÚC ĐỀ TÀI 17](#_Toc199868697)

[Chương 2. CƠ SỞ LÝ THUYẾT 18](#_Toc199868698)

[2.1 CÁC KHÁI NHIỆM, THUẬT NGỮ 18](#_Toc199868699)

[2.1.1 Hệ thống chấm bài tự động 18](#_Toc199868700)

[2.1.2 Cơ sở dữ liệu (CSDL) 19](#_Toc199868701)

[2.1.3 Mô hình 3 lớp 20](#_Toc199868702)

[2.2 CÔNG NGHỆ SỬ DỤNG 21](#_Toc199868703)

[2.2.1 API và RESTful API 21](#_Toc199868704)

[2.2.2 ReactJS + Vite và ChakraUI 24](#_Toc199868705)

[2.2.3 Docker 25](#_Toc199868706)

[2.2.4 Các công nghệ khác 28](#_Toc199868707)

[Chương 3. KẾT QUẢ NGHIÊN CỨU VÀ THẢO LUẬN 29](#_Toc199868708)

[3.1 KẾT QUẢ NGHIÊN CỨU 29](#_Toc199868709)

[3.1.2 Yêu cầu chức năng nghiệp vụ 31](#_Toc199868710)

[3.1.3 Mô tả chi tiết các đối tượng 33](#_Toc199868711)

[3.1.4 Thiết kế xây dựng API 45](#_Toc199868712)

[3.1.5 Thiết kế giao diện kết nối API 49](#_Toc199868713)

[3.2 THẢO LUẬN 74](#_Toc199868714)

[3.2.1 Đánh giá hiệu quả và ưu điểm của giải pháp 74](#_Toc199868715)

[3.2.2 Hạn chế của hệ thống hiện tại 76](#_Toc199868716)

[3.2.3 Đề xuất hướng phát triển trong tương lai 77](#_Toc199868717)

[Chương 4. KẾT LUẬN VÀ KIẾN LUẬN 79](#_Toc199868718)

[4.1 KẾT LUẬN 79](#_Toc199868719)

[4.2 KIẾN LUẬN 79](#_Toc199868720)

[TÀI LIỆU THAM KHẢO 81](#_Toc199868721)

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt, thuật ngữ** | **Giải nghĩa** |
|  | CSDL | Cơ sở dữ liệu |
|  | VPL | Virtual Programming Lab |
|  | DBMS | Database Management System  Hệ quản trị cơ sở dữ liệu |
|  | LMS | Learning Management System  Hệ thống quản lý học tập |
|  | API | Application Programing Interface |
|  | REST | Representation State Transfer |
|  | HTTP | Hypertext Transfer Protocol |
|  | SOAP | Simple Object Access Protocol |
|  | FE | Front-end |
|  | BE | Back-end |
|  | JWT | Json Web Token |

DANH MỤC HÌNH ẢNH

[Hình 2.1 Giao diện nộp bài ở NTUCoder 18](#_Toc199868629)

[Hình 2.2 Sơ đồ mô phỏng hệ thống chấm bài tự động 19](#_Toc199868630)

[Hình 2.3 Kiến trúc RESTful API 23](#_Toc199868631)

[Hình 2.4 Logo của Vite và React 24](#_Toc199868632)

[Hình 2.5 Logo biểu tượng của Docker 25](#_Toc199868633)

[Hình 2.6 Cấu trúc hoạt động của Docker 26](#_Toc199868634)

[Hình 3.1 Sơ đồ dữu liệu của Coder-LMS (Phần 1) 29](#_Toc199868635)

[Hình 3.2 Sơ đồ dữ liệu của Coder-LMS (Phần 2) 30](#_Toc199868636)

[Hình 3.3 Giao diện đăng nhập 49](#_Toc199868637)

[Hình 3.4 Kết quả trả về của API trên 50](#_Toc199868638)

[Hình 3.5 Giao diện đăng ký 50](#_Toc199868639)

[Hình 3.6 Giao diện lỗi khi bỏ trống 51](#_Toc199868640)

[Hình 3.7 Lỗi thiếu chữ in hoa 51](#_Toc199868641)

[Hình 3.8 Lỗi thiếu kí tự 51](#_Toc199868642)

[Hình 3.9 Lỗi thiếu chữ số 51](#_Toc199868643)

[Hình 3.10 Lỗi thiếu ký tự đặc biệt 52](#_Toc199868644)

[Hình 3.11 Giao diện danh sách các khóa học 52](#_Toc199868645)

[Hình 3.12 Giao diện ở chi tiết khóa học (bên trái) 53](#_Toc199868646)

[Hình 3.13 Tab giới thiệu về khóa học 54](#_Toc199868647)

[Hình 3.14 Tab danh sách chủ đề và bài học 55](#_Toc199868648)

[Hình 3.15 Tab đánh giá của người dùng 56](#_Toc199868649)

[Hình 3.16 Tab các bình luận của người dùng 57](#_Toc199868650)

[Hình 3.17 Người dùng trả lời bình luận của người khác 57](#_Toc199868651)

[Hình 3.18 Giao diện ở chi tiết khóa học (bên phải) 58](#_Toc199868652)

[Hình 3.19 Cảnh báo cần đăng nhập 58](#_Toc199868653)

[Hình 3.20 Sau khi bấm đăng ký 59](#_Toc199868654)

[Hình 3.21 Giao diện xác nhận hủy đăng ký khóa học 60](#_Toc199868655)

[Hình 3.22 Chi tiết bài học – phần mô tả 61](#_Toc199868656)

[Hình 3.23 Chi tiết bài học – phần giáo trình 62](#_Toc199868657)

[Hình 3.24 Danh sách các bài tập 63](#_Toc199868658)

[Hình 3.25 Chi tiết bài tập 63](#_Toc199868659)

[Hình 3.26 Code mẫu 63](#_Toc199868660)

[Hình 3.27 Testcase của bài tập 63](#_Toc199868661)

[Hình 3.28 Thông báo bị sai testcase 64](#_Toc199868662)

[Hình 3.29 Hiển thị lỗi cấu trúc code 64](#_Toc199868663)

[Hình 3.30 Khi chạy thử thành công 65](#_Toc199868664)

[Hình 3.31 Thông báo nộp bài thành công. 65](#_Toc199868665)

[Hình 3.32 Lịch sử nộp bài 66](#_Toc199868666)

[Hình 3.33 Đánh dấu hoàn thành bài tập 66](#_Toc199868667)

[Hình 3.34 Đánh dấu bài học đó hoàn thành 66](#_Toc199868668)

[Hình 3.35 Điểm trong trang cá nhân 67](#_Toc199868669)

[Hình 3.36 Hiển thị tiến trình hoàn thành 68](#_Toc199868670)

[Hình 3.37 Giao diện xếp hạng 68](#_Toc199868671)

[Hình 3.38 Modal tìm kiếm 69](#_Toc199868672)

[Hình 3.39 Phần gợi ý khi nhập từ khóa 70](#_Toc199868673)

[Hình 3.40 Giao diện Dashboard 71](#_Toc199868674)

[Hình 3.41 Giao diện index của khóa học 71](#_Toc199868675)

[Hình 3.42 Giao diện nhập khóa học mới 72](#_Toc199868676)

[Hình 3.43 Giao diện nhập của chủ đề 73](#_Toc199868677)

[Hình 3.44 Giao diện nhập của bài học 74](#_Toc199868678)

[Hình 3.45 Giao diện nhập của bài toán (phần 1) 74](#_Toc199868679)

[Hình 3.46 Giao diện nhập của bài toán (phần 2) 74](#_Toc199868680)

DANH MỤC BẢNG BIỂU

[Bảng 2. 1 Sự khác nhau giữa REST và SOAP 23](#_Toc199868604)

[Bảng 3. 1 Bảng mô tả chức năng nghiệp vụ 31](#_Toc199868605)

[Bảng 3. 2 Bảng Accounts 33](#_Toc199868606)

[Bảng 3. 3 Bảng Roles 34](#_Toc199868607)

[Bảng 3. 4 Bảng Coders 34](#_Toc199868608)

[Bảng 3. 5 Bảng Courses 35](#_Toc199868609)

[Bảng 3. 6 Bảng CourseCategories 36](#_Toc199868610)

[Bảng 3. 7 Bảng Topics 36](#_Toc199868611)

[Bảng 3. 8 Bảng Lessons 37](#_Toc199868612)

[Bảng 3. 9 Bảng LessonProblems 37](#_Toc199868613)

[Bảng 3. 10 Bảng Problems 38](#_Toc199868614)

[Bảng 3. 11 Bảng ProblemCategories 39](#_Toc199868615)

[Bảng 3. 12 Bảng Categories 39](#_Toc199868616)

[Bảng 3. 13 Bảng Submissions 39](#_Toc199868617)

[Bảng 3. 14 Bảng Compilers 40](#_Toc199868618)

[Bảng 3. 15 Bảng TestCases 41](#_Toc199868619)

[Bảng 3. 16 Bảng TestRuns 41](#_Toc199868620)

[Bảng 3. 17 Bảng Solved 42](#_Toc199868621)

[Bảng 3. 18 Bảng Matches 42](#_Toc199868622)

[Bảng 3. 19 Bảng Enrollments 43](#_Toc199868623)

[Bảng 3. 20 Bảng Badges 43](#_Toc199868624)

[Bảng 3. 21 Bảng Blogs 43](#_Toc199868625)

[Bảng 3. 22 Bảng Reviews 44](#_Toc199868626)

[Bảng 3. 23 Bảng Comments 44](#_Toc199868627)

[Bảng 3. 24 Bảng Progresses 45](#_Toc199868628)

PHẦN MỞ ĐẦU

1. Lý do chọn đề tài

Trong thời đại 4.0 hiện nay, nhu cầu cho việc lập trình, học tập lập trình ngày càng tăng cao và nhất là đối với sinh viên ngành công nghệ thông tin. Tuy nhiên, việc này đòi hỏi các khóa học lập trình cần một hệ thống quản lý học tập (LMS – Learning Management System) hiện đại, tiện lợi và có khả năng phát triển, mở rộng trong tương lai.

Bên cạnh đó, Trường Đại học Nha Trang (NTU) đang sử dụng hệ thống quản lý học tập **Moodle** kết hợp với công cụ chấm bài lập trình tự động **VPL (Virtual Programming Lab)**. Tuy nhiên, việc giảng dạy và học lập trình trên Moodle vẫn còn nhiều hạn chế do đây không phải là một nền tảng được thiết kế chuyên biệt cho môn học lập trình. Các chức năng như theo dõi tiến độ học tập, tổ chức bài học theo lộ trình rõ ràng hay đánh giá bài lập trình vẫn còn rời rạc và thiếu sự tối ưu. Song song đó, NTU cũng đã phát triển hệ thống NTUCoder – một nền tảng hỗ trợ sinh viên luyện tập lập trình và tham gia các kỳ thi thuật toán. Dù vậy, NTUCoder hiện chỉ mới cung cấp các bài tập đơn lẻ và các kỳ thi rời rạc, chưa phát triển thành một hệ thống học tập hoàn chỉnh theo khóa học. Điều này khiến giảng viên khó tổ chức các chương trình học lập trình bài bản, trong khi sinh viên cũng gặp khó khăn khi muốn theo đuổi một lộ trình học rõ ràng và có hệ thống.

Xuất phát từ thực trạng này, em quyết định thực hiện đề tài **“Xây dựng, nâng cấp website chấm bài lập trình thuật toán tự động NTUCoder (website LMS học lập trình)”** nhằm phát triển NTUCoder thành một nền tảng học lập trình toàn diện hơn. Hệ thống sẽ không chỉ giữ lại chức năng chấm bài và thực thi thuật toán mà còn tích hợp các yếu tố quản lý khóa học, hỗ trợ sinh viên đăng ký, theo dõi quá trình học và kết quả học tập của mình. Bên cạnh đó, việc bổ sung các tính năng như bài viết học thuật, phản hồi và đánh giá khóa học sẽ giúp nâng cao trải nghiệm người học, đồng thời hỗ trợ giảng viên trong công tác giảng dạy. Với đề tài này, em mong muốn mang đến một giải pháp thực tiễn giúp tự động hóa, minh bạch hóa và hệ thống hóa việc học lập trình tại NTU, tạo điều kiện để sinh viên phát triển kỹ năng lập trình một cách bài bản và hiệu quả hơn.

2. Mục tiêu đề tài

* **Hỗ trợ giảng viên:** Giảm bớt khối lượng công việc đánh giá bài tập.
* **Kiến thức:** giảng viên có thể đưa lên các khóa học từ chủ đề đến các bài học, bài tập liên quan để cho sinh viên có thể tự đăng ký vào học và từ đó giúp cho sinh viên có khả năng tự nghiên cứu cũng như tự nâng cao khả năng lập trình cho cá nhân.
* **Phát triển kỹ năng cho sinh viên:** Tạo ra một môi trường học tập thực tế, nơi sinh viên có thể liên tục rèn luyện và cải thiện kỹ năng lập trình thông qua phản hồi tự động và kịp thời.
* **Nâng cao hiệu quả giáo dục:** Bằng việc ứng dụng công nghệ tự động và kiểm thử toàn diện, NTU-Coder-LMS hướng tới việc nâng cao chất lượng giảng dạy và đánh giá, góp phần phát triển cộng đồng lập trình viên chất lượng cao.

3. Phương pháp nghiên cứu

* Khảo sát, tìm hiểu hiểu mô hình NTUCoder của trường Đại học Nha Trang.
* Phân tích, thiết kế hệ thống quản lý khóa học áp dụng cho trường Đại học Nha Trang.
* Tìm hiểu, nghiên cứu và áp dụng các kỹ thuật, mô hình lập trình, ngôn ngữ Javascript, C#, framework Reactjs + Vite, MySQL và framework .NET core 8 để triển khai hai phần chính của một trang web là Front-end và Back-end.

4. Bố cục trình bày

Nôi dung của đồ án gồm:

* Chương 1: Tổng quan về vấn đề nghiên cứu
* Chương 2: Cơ sở lý thuyết
* Chương 3: Kết quả nghiên cứu và thảo luận
* Chương 4: Kết luận và kiến luận

1. TỔNG QUAN VỀ WEBSITE NTU-CODER-LMS
   1. MÔ TẢ BÀI TOÁN

**NTU-CODER-LMS** là một hệ thống quản lý các khóa học liên quan đến lập trình và chấm bài lập trình tự động được xây dựng trên nền tảng sẵn có là NTUCoder. Website trước đây NTUCoder là một nền tảng liên quan đến hệ thống chấm lập trình tự động của trường Đại học Nha Trang và nền tảng này cho phép sinh viên tham gia vào các bài tập và các cuộc thi được giảng viên đưa ra. Tuy nhiên, qua quá trình khảo sát và trải nghiệm thực tế hệ thống vẫn còn tồn động một vài vấn đề liên quan đến giao diện trực quan đối với người dùng cũng như là giao diện dành cho giảng viên và quản trị. Giao diện hiện tại của NTUCoder vẫn còn quá đơn giản, thiếu tính thẩm mỹ và điều đó làm giảm đi sự hứng thú khi sinh viên vào để làm bài. Hệ thống vẫn chưa có xếp hạng cho các người lập trình khi giải các bài tập giống như LeetCode, CodeLearn, … được biết đến có giao diện hiện đại, hỗ trợ đa dạng ngôn ngữ lập trình và tính năng xếp hạng cho người lập trình theo thời gian thực.

Do đó, bài toán được đặt ra trong đề tài này là xây dựng và nâng cấp toàn diện hệ thống NTUCoder nhưng với lối đi hơi khác so với website cũ. Website mới sẽ liên quan đến quản lý khóa học và quản lý bài tập hơn là chỉ quản lý bài tập như là NTUCoder điều đó cũng giáp làm khắc phục hạn chế kiến thức khi lập trình mà phải đi tìm tài liệu ở nơi khác. Cụ thể là, hệ thống sẽ cung cấp giao diện hiện đại, thân thiện, hỗ trợ đầy đủ các ngôn ngữ lập trình phổ biến như C, C++, Java, Python, … Đồng thời, hệ thống phải cho phép giảng viên dễ dàng có thể tạo ra các khóa học, tạo các chủ đề, bài học cũng như là bài tập, thiết lập testcase, chấm điểm tự động, và thiết lập xếp hạng cho người cập nhật theo thời gian thực. Người dùng khi đăng ký khóa học thì cần phải đăng nhập và khi làm xong mỗi bài sẽ được 100 điểm cũng như phần trăm hoàn thành khóa học đó. Người dùng sẽ có một trang để theo dõi những tiến độ mà các khóa học họ đã đăng ký, hệ thống cũng phân quyền rõ ràng giữa người quản trị, giảng viên và sinh viên. Việc làm ra hệ thống đăng ký khóa học lập trình trên nền tảng chấm bài lập trình tự động NTU-CODER-LMS giúp cho người dùng có thể cải thiện trải nghiệm, nâng cao hiệu năng giảng dạy của giảng viên cũng như khả năng tự học tập của sinh viên. Ngoài ra, hệ thống này không chỉ giúp sinh viên mà còn giúp cho những người muốn bước vào con đường lập trình hay là những người đã quên kiến thức cũ nhằm thúc đẩy con đường học thuật trong cộng đồng lập trình.

* 1. XÂY DỰNG PHƯƠNG PHÁP VÀ NỘI DUNG NGHIÊN CỨU
     1. Phương pháp nghiên cứu

Trong quá trình thực hiện đề tài, em đã sử dụng nhiều phương pháp nghiên cứu để có thể đưa là các kết quả trực quan, đảm bảo tính khoa học và thực tiễn. Đầu tiên, phương pháp được sử dụng là khảo sát và thu thập các thông tin từ những người dùng thực tế là sinh viên và giảng viên của trường Đại học Nha Trang thông qua hình thức phỏng vấn trực tiếp, phiếu khảo sát Google Form, bảng hỏi. Nội dung khảo sát chủ yếu là đánh giá mức độ hài lòng đối với hệ thống NTUCoder hiện tại, các vấn đề cần sửa chữa, nhu cầu bổ sung chức năng, và mong muốn về giao diện người dùng. Từ đó đưa ra các yêu cầu cho hệ thống hiện tại để có thể xác định được hướng đi và cũng như cấu trúc các chức năng cần có. Trên nền tảng đó, quyết định là nâng cấp website cũ NTUCoder hay là phát triển ra một hệ thống mới.

* + 1. Nội dung nghiên cứu

Nội dung nghiên cứu của đề tài tập trung vào việc xây dựng và phát triển hệ thống quản lý học tập trực tuyến chuyên biệt dành riêng cho lĩnh vực lập trình – NTU-CODER-LMS. Cụ thể, đề tài được phát triển theo các nội dung chính như sau: khảo sát và phân tích hệ thống hiện tại, thiết kế lại kiến trúc giao diện và chức năng, lập trình nâng cấp hệ thống, và đánh giá kết quả sau cải tiến. Về công nghệ sử dụng, hệ thống được xây dựng với ngôn ngữ C# và ASP.NET Core API cho phần backend nhằm mục đích là tính ổn định, tải dữ liệu nhanh chóng và khả năng mở rộng. Riêng phần frontend sử dụng ngôn ngữ Javascript với framework Reactjs + Vite kết hợp với ChakraUI để cho giao diện trở nên đẹp mắt, hiện đại và phù hợp với mọi loại thiết bị. Cơ sở dữ liệu được sử dụng cho website này là MySQL, nhằm mục đích khả năng lưu trữ lớn và truy vấn hiệu quả. Trong quá trình thực hiện đề tài, em cũng đã áp dụng mô hình 3 lớp cho website của mình để phân tách rõ ràng giữa giao diện người dùng, quản lý dữ liệu và cũng như phần xử lý logic. Thêm vào đó, các chức năng chính được nghiên cứu và phát triển bao gồm: hệ thống tạo và đăng ký khóa học, thêm các bài toán vào các khóa học, bảng xếp hạng được reset theo thời gian thực, phân quyền giữa các giảng viên, sinh viên và quản trị. Việc tích hợp các công nghệ và phương pháp trên nhằm tạo ra một nền tảng học lập trình hiệu quả, hiện đại và phù hợp với yêu cầu đào tạo tại Trường Đại học Nha Trang.

* + 1. Lý do lựa chọn phương pháp nghiên cứu

Việc lựa chọn phương pháp nghiên cứu là để giải quyết vấn đề trong đề tài được cân nhắc dựa trên tính chất khoa học, phù hợp với mục tiêu, cũng như là tìm ra các dữ liệu liên quan đến phạm vi phát triển hệ thống NTUCoder hiện tại. Lựa chọn phương pháp khảo sát và thu thập các thông tin từ những người dùng thực tế là để có các thông tin dữ liệu chân thật, khách quan nhất về những nhu cầu và kỳ vọng đến từ thực tiễn của những người dùng - sinh viên, giảng viên là những đối tượng sử dụng trực tiếp của hệ thống. Điều này là để có thể phát hiện ra những khía cạnh điểm mạnh, điểm yếu và những nhu cầu cần cải thiện cho hệ thống. Để có thể hiểu rõ về cấu trúc cũng như là cách vận hành của một hệ thống chấm bài tự động, giao diện người dùng, các chức năng bị hạn chế, đó là lý do cho việc lựa chọn phương pháp phân tích – từ đó có thể cấu trúc lại hệ thống, nâng cấp hoặc mở rộng hệ thống và rồi đưa ra những giải pháp để có thể giải quyết những vấn đề mà hệ thống cũ còn đang mắc phải.

Phương pháp nghiên cứu thực nghiệm với sự hỗ trợ của những công cụ mạnh mẽ, thư viện mã nguồn mở (như Docker, API, ReactJS, Node.js) điều này làm cho việc phát triển hệ thống tiết kiệm thời gian, đồng thời thì việc này cũng giúp cho hệ thống về tương lại dễ dàng trong việc mở rộng và bảo trì. Giao diện sử dụng những thư viện sẵn có như ChakraUI ngoài việc tiết kiệm thời gian thì cũng giúp em có thể thiết kế những giao diện theo ý muốn người dùng một cách dễ dàng hơn, còn có nâng cao hiệu suất khi những thứ dư thừa trong lập trình đã được loại bỏ bởi những người tạo ra ChakraUI, cũng như dễ dàng thiết lập cho những tính năng thích ứng với mọi loại thiết bị và giao diện thân thiện, hiện đại phù hợp với xu thế của hiện tại. Ngoài ra, để tối ưu quy trình phát triển thì em đã dùng mô hình linh hoạt với kiến trúc 3 lớp – điều này cũng là giúp cho việc phát trình trở nên rành mạch rõ ràng hơn, việc phân tách backend và frontend ra 2 lớp khác nhau cũng giúp cho việc sau này khi làm nhóm cũng dễ dàng hơn và có thể thuận tiện bảo trì và mở rộng trong tương lai gần.

* 1. KHẢO SÁT, ĐÁNH GIÁ THỰC TRẠNG VÀ TRẢI NGHIỆM LIÊN QUAN ĐẾN HỆ THỐNG NGHIÊN CỨU
     1. Khảo sát trải nghiệm người dùng đối với trang website NTUCoder

Nhằm đánh giá mức độ hài lòng và khó khăn mà người dùng, cụ thể ở đây là giảng viên và sinh viên thuộc trường Đại học Nha Trang trong quá trình trải nghiệm các chức năng và sự vận hành của hệ thống chấm bài tự động, đồng thời kết hợp với kinh kinh nghiệm cá nhân trong quá trình sử dụng hệ thống trong các học phần lập trình tại Trường Đại học Nha Trang. Qua đó cho thấy được những kết quả khác nhau như sau:

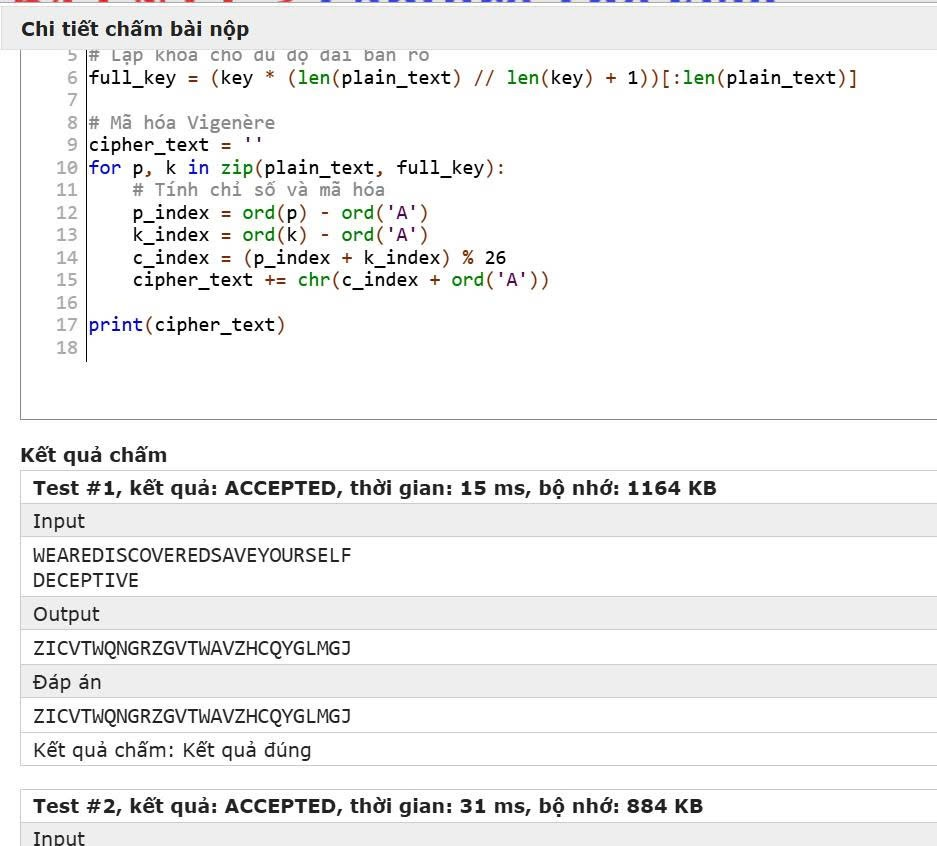
* Ưu điểm:
* Hệ thống đã cung cấp kho bài tập lập trình khá phong phú, có xếp loại cho những bài tập đó theo các mức độ khó – trung bình – dễ.
* Giao diện đơn giản, dễ dàng cho người mới lần đầu sử dụng.
* Có nhiều ngôn ngữ lập trình như C/C++, pascal, python, java 8.
* Chức năng chấm điểm tự động giúp sinh viên có thể kiểm tra kết quả và cải thiện bài làm nhanh chóng.
* Hạn chế:
* Giao diện chưa được đẹp, hiện đại và tối ưu trên điện thoại.
* Kết quả chấm bài đôi khi chỉ thông báo “Wrong Answer” mà không hiển thị chi tiết input/output gây lỗi, khiến sinh viên khó xác định sai sót.
* Không có thống kê tiến độ làm bài, lịch sử nộp bài, hay đánh giá mức độ hoàn thành của cá nhân so với lớp học.
* Không có những lý thuyết liên quan tới bài tập đó khi người dùng cần tới khi đang làm.
  + 1. Trải nghiệm cảm giác quản lý trang website NTUCoder

Bên cạnh việc khảo sát và phân tích góc nhìn của các giảng viên và sinh viên, em cũng đã tiến hành đánh giá chủ quan dựa trên quá trình trải nghiệm trực tiếp hệ thống NTUCoder trong vai trò người quản trị. Ở vai trò này, em đã tiến hành thử nghiệm một vài chức năng chính như thêm xóa sửa các bài tập, tổ chức kỳ thi, xem kết quả, và theo dõi hoạt động của người dùng. Qua quá trình trải nghiệm, em thấy giao diện của hệ thống còn quá đơn giản, thiếu tính trực quan và chưa đáp ứng được những nhu cầu thao tác phức tạp trong quá trình vận hành hệ thống. Một số chức năng chính như là thống kê kết quả học, quản lý bài tập theo loại bài tập, và phân quyền người dùng chưa được triển khai hoặc chưa đầy đủ, gây khó khăn trong việc vận hành hệ thống ở quy mô lớn. Việc tiếp cận và sử dụng hệ thống trong vai trò quản trị viên đã giúp em nhận diện rõ hơn các điểm còn hạn chế của hệ thống, từ đó đề xuất các giải pháp nâng cấp có tính thực tiễn cao. Những đánh giá này đóng vai trò quan trọng trong việc xác định định hướng cải tiến, phù hợp với yêu cầu sử dụng hệ thống trong môi trường đào tạo lập trình quy mô lớn tại nhà trường.

* 1. CẤU TRÚC ĐỀ TÀI
* **Chương 1: Tổng quan về website NTU-Coder-LMS** – trình bày mô tả bài toán, phương pháp, nội dung, và lý do chọn lựa nghiên cứu các dữ liệu từ hệ thống cũ NTUCoder để phát triển hệ thống mới.
* **Chương 2: Cơ sở lý thuyết** – trình bày ngắn gọn khái niệm, thuật ngữ, công nghệ và cơ sở lý thuyết có liên quan mật thiết đến sự phát triển cho hệ thống mới. Phân tích chức năng theo yêu cầu, hệ thống LMS, hệ thống chấm bài tự động, API, các công cụ hỗ trợ backend/frontend và thiết kế kiến trúc hệ thống theo mô hình 3 lớp.
* **Chương 3: Kết quả nghiên cứu –** trình bày nội dung liên quan đến quá trình thực hiện nghiên cứu, quá trình frontend kết nối với backend, CSDL, giao diện quản lý, giao diện người dùng, các chức năng chính và kiểm thử trước khi đưa vào hoạt động.
* **Chương 4: Kết luận và kiến nghị -** trình bày về kết quả sau khi thực hiện đề tài và từ đó đưa ra những phân tích về sản phẩm mới đối với sản phẩm cũ. Đưa hướng đi mở rộng cho sản phẩm ở tương lai.

1. CƠ SỞ LÝ THUYẾT
   1. CÁC KHÁI NHIỆM, THUẬT NGỮ
      1. Hệ thống chấm bài tự động

Hệ thống chấm bài tự động là một nền tảng phần mềm có chức năng tự động biên dịch, thực thi và đánh giá kết quả của nội dung mã nguồn mà người lập trình nộp lên, dựa trên dữ liệu của **bộ kiểm thử** (**testcase)** mà giảng viên hay là quản trị viên cung cấp sẵn, ngoài ra cũng gồm các dữ liệu đầu vào và kết quả đầu ra để kiểm tra độ chính xác của lời giải. Mục tiêu của hệ thống này là cung cấp một quy trình đánh giá việc tự động đánh giá bài lập trình một cách **nhanh chóng, khách quan và nhất quán**, thay thế cho việc chấm bài thủ công truyền thống. Dưới đây là hình ảnh mô phỏng cho việc nộp bài trên website NTUCoder:

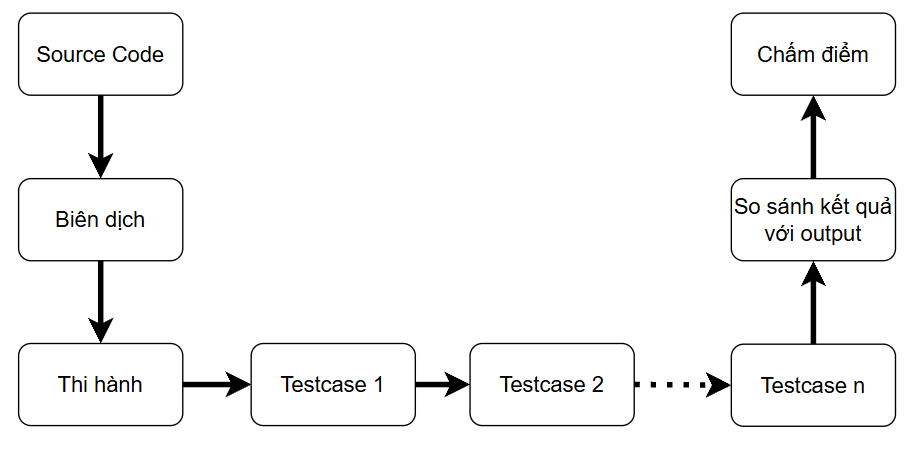


Giao diện nộp bài ở NTUCoder

Hệ thống chấm bài theo phương pháp “hộp đen” (Black-box testing) đây là phương pháp chấm bài mà trong đó hệ thống không quan tâm đến cách người dùng (hay là người lập trình) thực hiện lời giải thế nào (cụ thể là code như thế nào), mà nó chỉ quan tâm đến đầu vào và kết quả đầu ra. Cụ thể là như sau:

* Mỗi lần chương trình chạy sẽ là chạy với một tập hợp testcase được thiết lập từ trước (input và output).
* Sau đó hệ thống sẽ so sánh kết quả thực thi của người dùng với kết quả mong đợi của từng testcase. Nếu tất cả đều khớp hết thì kết quả là chính xác.
* Việc chấm điểm có thể thực hiện theo dạng **đúng/sai tuyệt đối** hoặc **chấm điểm từng phần**, tùy thuộc vào cấu hình hệ thống.

Cách này giúp đảm bảo tính **khách quan**, **nhất quán** và **tự động hóa hoàn toàn** quy trình đánh giá. Dưới đây là sơ đồ mô phỏng quy trình hoạt động của hệ thống chấm bài tự động với phương pháp hộp đen:



Sơ đồ mô phỏng hệ thống chấm bài tự động

* + 1. Cơ sở dữ liệu (CSDL)

**Cơ sở dữ liệu (database)** là một tập hợp những dữ liệu điện tử tổ chức có cấu trúc, được lưu trữ trên các thiết bị trữ tin, và thêm vào đó còn có chọn lọc cho nhiều người sử dụng với những mục đích khác nhau. **Dữ liệu trong CSDL** thường được lưu dưới dạng bảng, trong đó thì mỗi hàng là một bản ghi và mỗi cột thì thuộc về thuộc tính, điều này giúp cho người dùng có thể dễ dàng thao tác và truy vấn các thông tin cần thiết.

Để có thể lưu trữ, truy vấn hoặc là chỉnh sửa dữ liệu một cách an toàn và hiệu quả thì ta cần dùng phần mềm **Hệ quản trị dữ liệu (DBMS)** ví dụ như là MySQL – đang được sử dụng trong bài đang làm, ngoài ra còn có những ví dụ như là SQL Server Manager của Microsoft, MongoDB, SQLite, …

Trong khuôn khổ đề tài em đang làm, hệ thống NTU-Coder-LMS đang sử dụng MySQL làm hệ quản trị CSDL chính. MySQL đóng vai trò là nơi lưu trữ các dữ liệu liên quan đến các thông tin về khóa học, bài tập, người dùng (giảng viên, sinh viên) và còn lưu trữ các thông tin khác như là tiến độ hoàn thành của một khóa học của một sinh viên, danh sách bài viết. Chẳng hạn như, khi một người dùng muốn có thể vào làm bài tập hoặc xem nội dung của khóa học thì họ cần phải đăng ký – lúc này MySQL có vai trò lưu trữ người dùng đã đăng ký vào hệ thống để giảng viên hay là quản trị viên có thể xem thấy được. Ngoài ra, khi trong DBMS có dữ liệu đã đăng ký thì người dùng mới có thể lập trình và nộp bài, sau khi nộp bài hệ thống sẽ tự chấm bài và trả về cho người nội dụng kết quả mà trước đó đã được lưu trước vào trong database. Nhờ vậy, việc thống kê, theo dõi kết quả, hoặc là đánh giá tiến độ cũng trở nên tiện lợi và dễ dàng lấy ra để hiển thị hơn.

* + 1. Mô hình 3 lớp

Mô hình 3 lớp (Three-Layer Architecture) là một dạng kiến trúc phổ biến, trong đó ứng dụng sẽ được chia làm ba lớp riêng biệt: **Lớp trình diễn (Presentation Layer), Lớp xử lý nghiệp vụ (Business Logic Layer) và Lớp truy cập dữ liệu (Data Access Layer).** Mỗi lớp sẽ đóng một vai trò cụ thể, giúp tổ chức và quản lý mã nguồn một cách hợp lý, rõ ràng, linh hoạt, dễ mở rộng, phát triển và dễ dàng bảo trì.

* **Lớp trình diễn (Presentation Layer)**: Đây là lớp giao tiếp trực tiếp với người dùng, có trách nhiệm đóng vai trò hiển thị giao diện và tiếp nhận đầu vào. Lớp này thường sử dụng những công nghệ như là HTML, CSS, JavaScript trong các ứng dụng web.
* **Lớp xử lý nghiệp vụ (Business Logic Layer):** Lớp này xử lý các bước logic, xử lý dữ liệu nhận được từ lớp trình diễn và thực hiện các quy tắc, các thuật toán cần thiết trước khi chuyển dữ liệu đã được xử lý qua lớp truy cập dữ liệu hoặc là trả về lớp trình diễn những kết quả được xử lý khi lấy từ lớp truy cập dữ liệu.
* **Lớp truy cập dữ liệu (Data Access Layer):** Còn lớp này chịu trách nhiệm tương tác với hệ quản trị cơ sở dữ liệu (DBMS), thực hiện các chức năng truy vấn, thêm, xóa, sửa dữ liệu. Lý do cho việc tách lớp này là để có thể dễ dàng quản lý dữ liệu và mang tính bảo mật hơn.
  1. CÔNG NGHỆ SỬ DỤNG
     1. API và RESTful API
        1. Khái niệm API

API (Application Programing Interface) là giao diện ứng dụng lập trình – là một tập hợp những quy tắc và giao thức cho phép các ứng dụng phần mềm giao tiếp với nhau. API gồm nhiều phương thức có quy tắc để cho các ứng dụng bên ngoài có thể giao tiếp với hệ thống để thực hiện các tương tác tính toán và trao đổi thông tin. Giúp các nhà lập trình phát triển tạo ra những service để cho những người lập trình khác có thể từ đó mà tận dụng mà không phải đi lập trình lại hay phải tải về.

* + - 1. Khái niệm RESTful API

REST (Representation State Transfer) là một giải pháp giao tiếp với ứng dụng web của Web Service. REST được giới thiệu bởi Roy Fielding vào năm 2000 và đã trở thành một chuẩn phổ biến trong việc triển khai xây dựng các dịch vụ về web. RESTful API là một kiểu thiết kế cho API tuân thủ theo các nguyên tắc của kiến trúc REST và sử dụng phương thức HTTP để giao tiếp giữa client và server. Các nguyên tắc chính của RESTful API là:

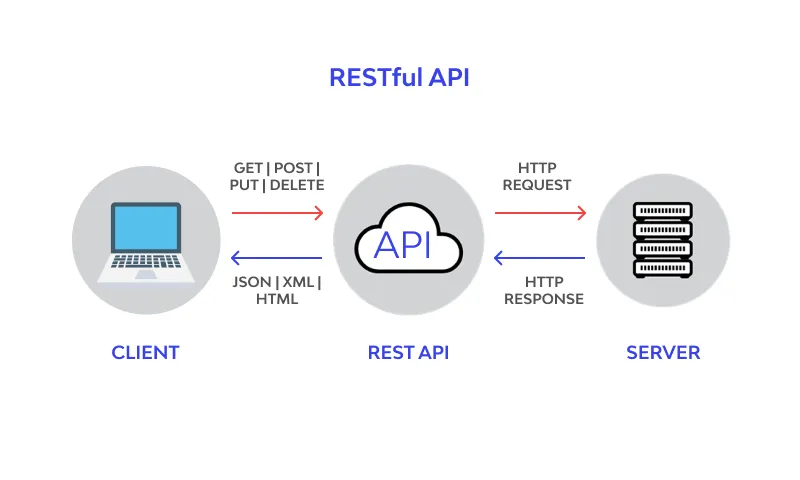
* **Client-Server**: Ở nguyên tắc này RESTful đã phân tách ra hai thành phần chính là **client (người dùng)** và **server (máy chủ)**, điều này giúp cực kỳ tốt cho việc cải thiện và mở rộng ứng dụng cho tương lai.
* **Stateless:** Server có thể hiểu và xử lý thông tin đến từ các yêu cầu được gửi từ client, mà không cần phải lưu lại trạng thái xử lý từ những yêu cầu trước đó.
* **Cacheable:** Phản hồi từ server có thể được lưu vào bộ nhớ cache để cải thiện hiệu suất và giảm tải cho server.
* **Giao diện thống nhất:** Sử dụng các phương thức của HTTP tiêu chuẩn như GET, POST, PUT, DELETE.
* **Hệ thống phân tầng:** Kiến trúc này là để đảm bảo việc mở rộng và bảo mật – đến từ việc sử dụng các thành phần trung gian như proxy, gateway.
* **Mã trên yêu cầu (Code on Demand – tùy chọn):** Server có thể cung cấp mã thực thi client khi cần thiết, chẳng hạn như là JavaScript.

Các phương thức HTTP thường được sử dụng trong việc lập trình web là:

* **GET**: Lấy thông tin từ tài nguyên.
* **POST**: Tạo mới một tài nguyên.
* **PUT**: Cập nhật toàn bộ thông tin một tài nguyên.
* **PATCH**: Cập nhật một phần thông tin.
* **DELETE**: Xóa một tài nguyên.

Ưu điểm có thể kể đến của việc sử dụng kiến trúc RESTful API:

* Tính tiện dụng: client có thể gọi và hiển thị ra các thông tin của một tài nguyên thông qua việc sử dụng URL, điều này cũng giúp cho các nhà lập trình dễ dàng hiểu và sử dụng.
* Dễ mở rộng bởi vì RESTful API không lưu lại trạng thái và việc giao tiếp của client – server là độc lập.
* Tương tác với nhiều định dạng dữ liệu: RESTful API hỗ trợ nhiều định dạng dữ liệu như JSON, XML, HTML, v.v., nhưng JSON là định dạng phổ biến nhất hiện nay vì tính nhẹ và dễ xử lý.
* Tính định địa chỉ: mỗi nguồn tài nguyên sẽ được có một địa chỉ URL riêng và lập trình viên có thể lấy toàn bộ thông tin cần thiết thông qua kết nối giữa các tài nguyên.



Kiến trúc RESTful API

So sánh sự khác nhau giữa REST và SOAP:

* + - * 1. Sự khác nhau giữa REST và SOAP

|  |  |  |
| --- | --- | --- |
| **STT** | **REST** | **SOAP** |
| 1 | REST là một cách thiết kế kiến trúc | SOAP là một giao thức |
| 2 | REST viết tắt của Representational State Transfer | SOAP là từ viết tắt của Simple Object Access Protocol (Giao thức truy cập đối tượng đơn giản) |
| 3 | REST có thể dùng các Web Services sử dụng SOAP vì nó có thể dùng bất kỳ giao thức nào như HTTP, SOAP | SOAP không thể sử dụng REST vì nó là một giao thức |
| 4 | REST sử dụng đỉa chỉ URI để cung cấp các dịch vụ | SOAP cung cấp các giao diện dịch vụ (services interfaces) cho các thành phần bên ngoài sử dụng |
| 5 | REST không định nghĩa nhiều chuẩn như SOAP | SOAP định nghĩa các chuẩn và quy tắc chặt chẽ |
| 6 | REST sử dụng băng thông và tài nguyên ít hơn SOAP | SOAP sử dụng băng thông và tài nguyên nhiều hơn REST |
| 7 | RESTful kế thừa chuẩn bảo mật tầng vận tải của giao thức mạng | SOAP định nghĩa chuẩn bảo mật của riêng nó |
| 8 | Ít tin cậy hơn – chẳng hạn, HTTP DELETE có thể trả về trạng thái OK ngay cả khi tài nguyên không được xóa | Tin cậy hơn |
| 9 | Thường không được dùng trong môi trường tính toán phân tán | Được thiết kế để dùng trong tính toán phân tán |
| 10 | REST hỗ trợ các định dạng dữ liệu khác nhau như text, HTML, XML, JSON | SOAP chỉ hỗ trợ định dạng dữ liệu XML |

* + 1. ReactJS + Vite và ChakraUI



Logo của Vite và React

ReactJS là một thư viện javascript được tạo bởi cộng tác giữa Facebook và Instagram. Nó cho phép lập trình viên thiết kế giao diện web một cách nhanh chóng. Phần hiển thị của React thường sử dụng các component mà chứa các component cụ thể hoặc là sử dụng các thẻ HTML như bình thường. Đặc biệt là một trong những đặc trưng của ReactJS là có thể render dữ liệu không những ở server mà còn ở client.

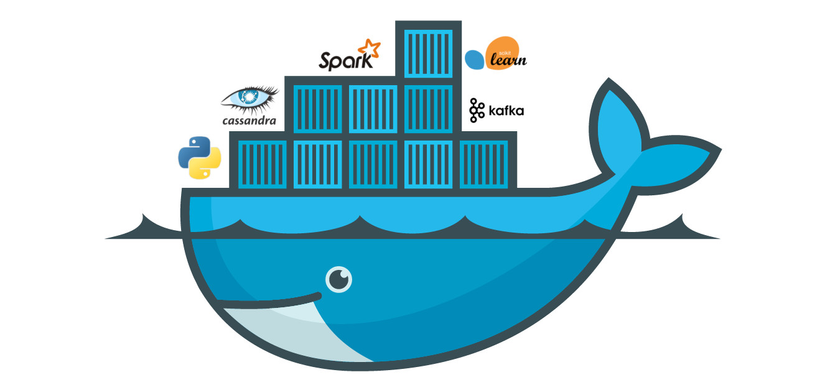
Đặc điểm nổi bật của ReactJS:

* Component-based: React chia các thành phần ra từng phần nhỏ giúp cho việc bảo trì, mở rộng và tái sử dụng trở nên dễ dàng hơn.
* Virtual DOM: sử dụng DOM để ảo hóa mỗi khi cập nhật giao diện.
* One-way Data Binding: Dữ liệu được truyền theo một chiều, giúp cho lập trình viên dễ dàng kiểm soát và debug.
* Hỗ trợ SSR (Server-Side Rendering): Giúp cải thiện hiệu suất và cải thiện CEO của ứng dụng.

Vite là công cụ build và phát triển ứng dụng web hiện đại. Ưu điểm của nó là build và khởi động ứng dụng nhanh. Cho phép cập nhật dữ liệu khi sửa đổi mà không cần phải tải lại toàn bộ trang.

ChakraUI là một thư viện thành phần UI dành cho React, cho phép chúng ta sử dụng các component có sẵn. Thư viện này cung cấp những component giúp cho lập trình viên dễ dàng sử dụng và tùy chỉnh. Ngoài ra, nó còn hỗ trợ theming cho phép chúng ta dễ dàng thay đổi giao diện theo chủ đề của ứng dụng và còn hỗ trợ responsive thích ứng với mọi thiết bị.

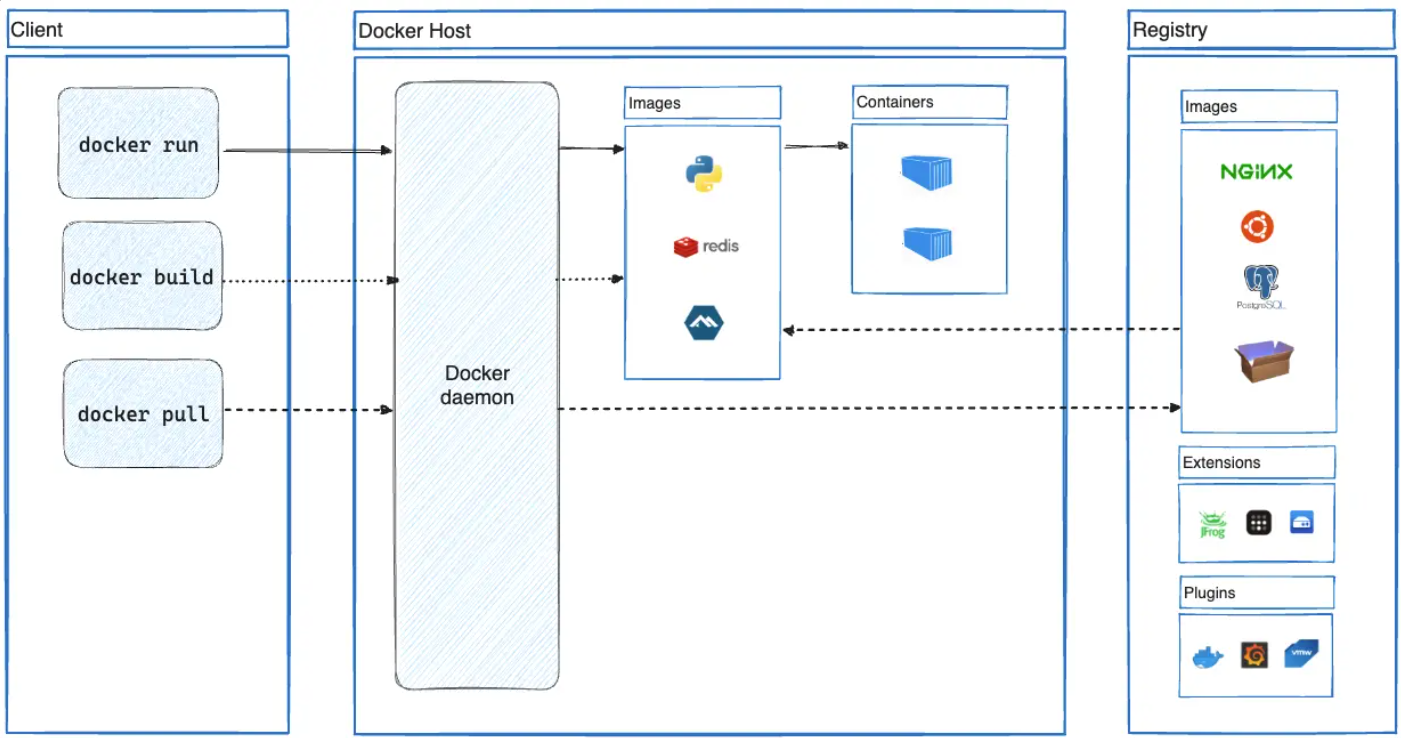
* + 1. Docker



Logo biểu tượng của Docker

Docker là một nền tảng mở để phát triển, vận chuyển và chạy ứng dụng. Nó tách biệt các ứng dụng của dự án ra khỏi cơ sở hạ tầng để ta có thể phân phối phần mềm nhanh chóng. Thêm vào đó, Docker cho phép chúng ta đóng gói ứng dụng và toàn bộ các thành phần phụ thuộc vào container – là một môi trường cách biệt. Nhờ có vậy, lập trình viên có thể triển khai ứng dụng phù hợp với mọi môi trường, bất kể máy chủ hay là hệ điều hành gì.

Dựa trên những tính năng đó mà đề tài của em đã triển khai dùng Docker để host khi ở localhost, nó cho phép em host database, và các ứng dụng lưu trữ khác thông qua image – container – volume mà không cần phải dùng đến phần mềm khác. Ngoài ra, nó còn có thể thực thi mã nguồn để được hiện chức năng chấm bài lập trình tự động. Docker có thể thực thi mã nguồn của nhiều ngôn ngữ khác nhau và trả về các kết quả tương ứng ví dụ như Python, C/C++, Java, … Lý do sử dụng Docker là để đảm bảo về các vấn đề trước khi deloy lên các server tránh các vấn đề lỗi không đúng biến, bảo mật và mang tính nhất quán. Để biết thêm về cấu trúc hoạt động của Docker xem **hình 2.3** dưới đây:



Cấu trúc hoạt động của Docker

Với quy trình chấm bài tự động sử dụng Docker đầu tiên ta phải tải image liên quan đến trình biên dịch code mà ta cần, ở trong hệ thống này hiện tại chỉ biên dịch C/C++ nên tải Image C++ thông qua câu lệnh “**docker pull gcc:12**” và sau đó ở phần BE thì gồm những bước sau:

* Bước 1: Lấy thông tin bài nộp và testcase

Khi người dùng bấm nộp bài (tức là submission), hệ thống sẽ bắt đầu truy vấn đến cơ sở dữ liệu để lấy:

* Mã nguồn của người dùng (SubmissonCode).
* Loại trình biên dịch (Compiler), ví dụ C++.
* Danh sách các testcase tương ứng với bài toán, mỗi testcase sẽ bao gồm **Input (Đầu vào)**  và **Expected Ouput (Đầu ra mong đợi).**
* Bước 2: Sinh lệnh Docker để biên dịch và thực thi

Đối với hệ thống khác sẽ tùy vào ngôn ngữ mà hệ thống Docker sẽ sinh ra mỗi câu lệnh tương ứng. Nhưng ở hệ thống đang làm của NTU-Coder-LMS thì chỉ C/C++. Để thực hiện được thì thực hiện những quy trình sau:

* Mã nguồn sẽ được mã hóa bằng Base64 để tránh những lỗi liên quan đến ký tự khi truyền vào lệnh shell.
* Tạo file mã nguồn ngay bên trong container từ chuỗi đã mã hóa.
* Tiến hành biên dịch (nếu cần) và thực thi chương trình với input tương ứng.

Ví dụ, đối với ngôn ngữ C++, lệnh Docker sẽ có dạng:

**docker run --rm --name code\_runner\_<id> gcc:12 sh -c "**

**echo '<base64\_code>' | base64 -d > main.cpp &&**

**g++ main.cpp -o main.out &&**

**echo '<input>' | ./main.out"**

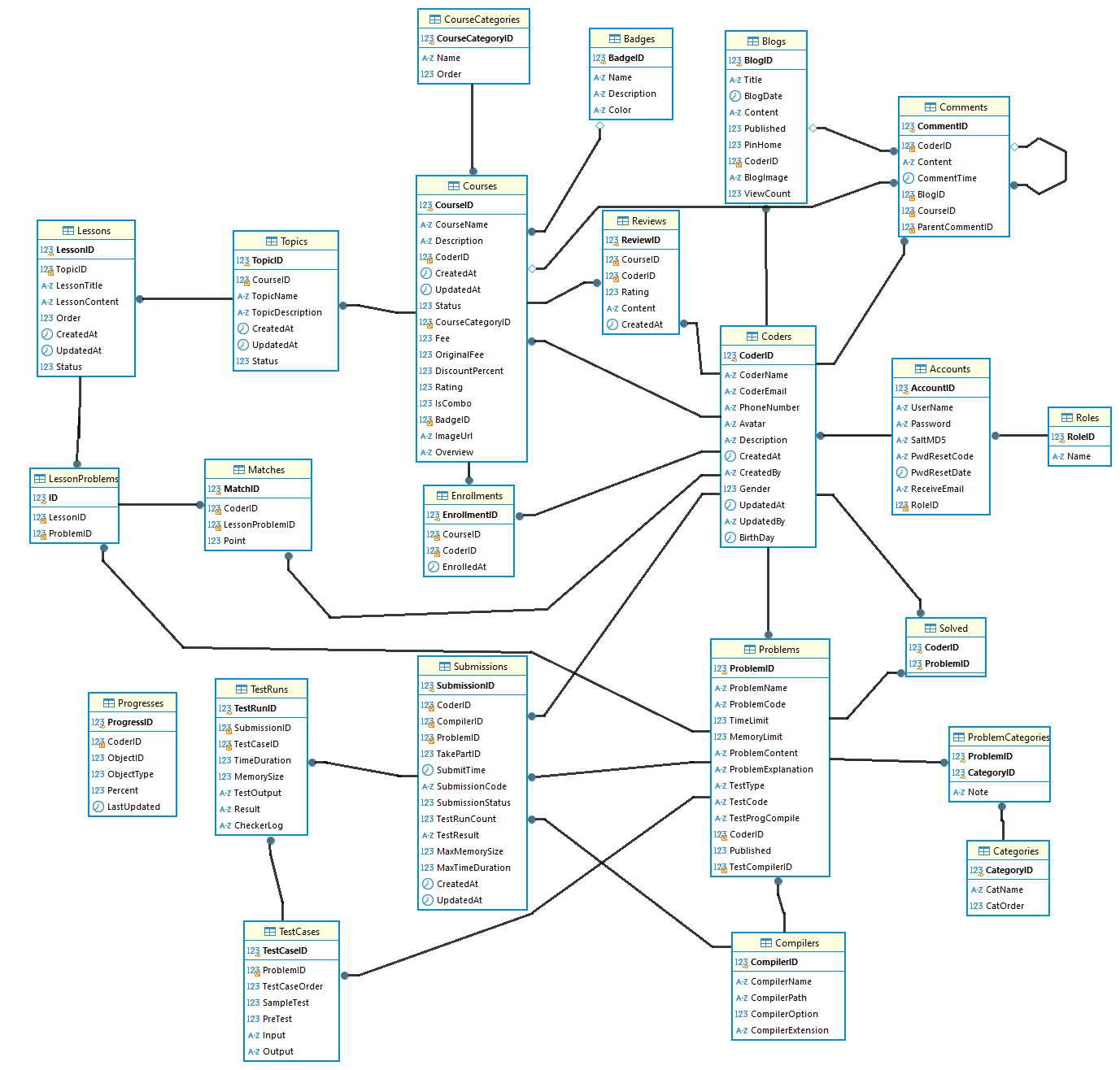
* Bước 3: Thực thi và thu thập kết quả
* Lệnh Docker được thực thi thông qua **cmd.exe** sử dụng lớp **ProcessStartInfo** trong C#.
* Hệ thống ghi nhận:
  + **StandardOutput**: đầu ra chương trình.
  + **StandardError**: thông báo lỗi nếu có.
  + Thời gian thực thi.
* Nếu chương trình chạy vượt quá giới hạn thời gian, hệ thống sẽ hủy container và trả về lỗi “Time Limit Exceeded”.
* **Bước 4: So sánh kết quả và chấm điểm**
* Nếu chương trình thực thi thành công, hệ thống sẽ so sánh đầu ra thực tế với đầu ra mong đợi của test case.
* Kết quả của mỗi test case được xác định là:
  + **Accepted**: nếu khớp hoàn toàn.
  + **Wrong Answer**: nếu khác nhau.
  + **Runtime Error** hoặc **Compilation Error**: nếu gặp lỗi khi chạy hoặc biên dịch.
* Kết quả cùng với thời gian chạy, bộ nhớ và log chi tiết được lưu lại vào bảng TestRun.
* **Bước 5: Cập nhật trạng thái bài nộp**

Sau khi thực thi toàn bộ test case, hệ thống cập nhật trạng thái tổng thể cho bài nộp, bao gồm:

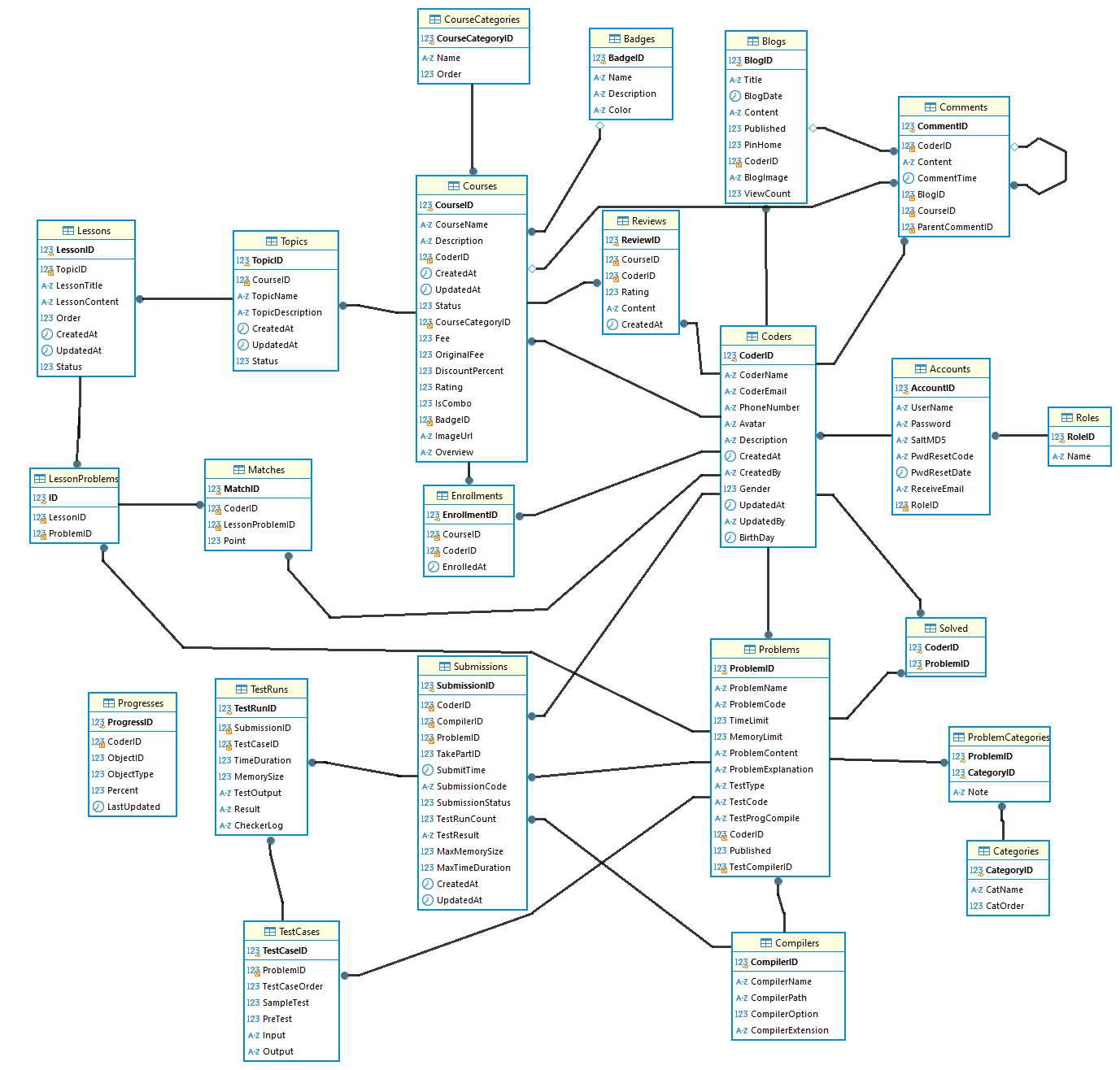
* Tổng số test case vượt qua.
* Trạng thái cuối cùng: **Accepted, Wrong Answer, Runtime Error**, v.v.
  + 1. Các công nghệ khác

Ngoài những công nghệ ở trên, em còn sử dụng MinIO để lưu trữ những hình ảnh, ví dụ như là avatar, ảnh bài viết và ảnh khóa học. MinIO giống như AWS S3 là một object storage server được implement những public API. Ngoài ra còn có Google Authentication là một công nghệ cho phép chúng ta dùng để đăng nhập bằng chính tài khoản Google mà không cần phải đăng ký. Dự án còn có sử dụng JWT dùng để xác thực người dùng bằng mã token.

1. KẾT QUẢ NGHIÊN CỨU VÀ THẢO LUẬN
   1. KẾT QUẢ NGHIÊN CỨU



Sơ đồ dữu liệu của Coder-LMS (Phần 1)



Sơ đồ dữ liệu của Coder-LMS (Phần 2)

* + 1. Yêu cầu chức năng nghiệp vụ

Bảng sau đây sẽ trình bày các chức năng nghiệp vụ chính trong hệ thống NTU-CODER-LMS, mô tả những chức năng, đối tượng sử dụng và kết quả mong đợi.

* + - * 1. Bảng mô tả chức năng nghiệp vụ

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên chức năng** | **Mô tả nghiệp vụ** | **Đối tượng sử dụng** |
|  | Đăng ký tài khoản | Người dùng nhập các thông tin cần thiết để đăng ký tài khoản mới | Người học |
|  | Đăng nhập | Người dùng sử dụng tên đăng nhập và mật khẩu để đăng nhập hoặc dùng tài khoản của google | Tất cả mọi người |
|  | Quên mật khẩu | Hệ thống sẽ gửi đến email đã được nhập một mã code sau đó người dùng phải xác nhận code đó và nhập mật khẩu mới | Người học |
|  | Phân quyền người dùng | Người dùng được phân quyền như sau: user, teacher, admin | Quản trị viên |
|  | Quản lý khóa học | Xóa, cập nhật thông tin và ẩn/hiển khóa học | Quản trị viên và giảng viên |
|  | Tạo khóa học | Nhập tên khóa học, ảnh đại diện, lựa chọn loại khóa học, mô tả và giới thiệu chung về khóa học | Quản trị viên và giảng viên |
|  | Quản lý chủ đề | Xóa, cập nhật thông tin và ẩn/hiển chủ đề | Quản trị viên và giảng viên |
|  | Tạo chủ đề | * Chọn khóa học, nhập tên chủ đề, mô tả và trạng thái * Có thể tạo một lúc nhiều chủ đề thuộc về một khóa học | Quản trị viên và giảng viên |
|  | Quản lý bài tập | Xóa, cập nhật thông tin và ẩn/hiển bài học  Hiển thị số lượng testcase | Quản trị viên và giảng viên |
|  | Thêm bài tập | * Chọn bài học, chọn trình biên dịch, nội dung bài tập và các thông tin khác. * Có thể tạo cùng lúc nhiều bài tập thuộc nhiều bài học khác nhau | Quản trị viên và giảng viên |
|  | Quản lý testcase | Thêm, xóa và cập nhật testcase | Quản trị viên và giảng viên |
|  | Quản lý người dùng | Thêm, xóa và cập nhật người dùng | Quản trị viên |
|  | Cập nhật hồ sơ cá nhân | Người dùng cập nhật avatar, tên, ngày sinh, số điện thoại và giới thiệu (yêu cầu đăng nhập) | Tất cả mọi người |
|  | Đăng ký khóa học | Người dùng cần phải đăng nhập mới có thể đăng ký khóa học và cũng có thể hủy khóa học | Người học |
|  | Chi tiết khóa học | Người dùng sẽ thấy các phần sau: Giới thiệu, giáo trình, đánh giá và bình luận | Người học |
|  | Đánh giá | Người dùng có thể đánh giá chất lượng khóa học và để lại bình luận (yêu cầu đăng nhập) | Người học |
|  | Bình luận | Người dùng với nhau có thể trao đổi qua lại và giảng viên cũng có thể | Người học và giảng viên |
|  | |  | | --- | |  | | Làm bài tập / Nộp bài | | | |  |  |  |  | | --- | --- | --- | --- | | |  | | --- | | Người học viết mã trong trình biên dịch, nộp bài để hệ thống chấm điểm tự động thông qua testcase |  |  | | --- | |  | | | |  | | Người học |
|  | Lịch sử làm bài | Người học sẽ xem những lần nộp vào thời gian nào, kết quả và điểm số đã làm | Người học |
|  | Thống kê tiến độ học | Hiển thị số bài đã đăng ký, đã hoàn thành hoặc hoàn thành bao nhiêu phần trăm | Tất cả mọi người |
|  | Tìm kiếm | Tìm kiếm theo 3 loại: khóa học, người dùng và bài viết, tìm theo từ khóa | Tất cả mọi người |
|  | Thêm mới bài viết | Nhập tiêu đề, ảnh đại diện và nội dung bài viết (yêu cầu đăng nhập) | Tất cả mọi người |

* + 1. Mô tả chi tiết các đối tượng
       1. Accounts

Đây là bảng ghi lưu lại những tài khoản đã được tạo bởi người dùng. Nó có chức năng cho phép người dùng lưu trữ thông tin tài khoản, thay đổi đổi mật khẩu, nhận mã code khi quên mật khẩu. Ngoài ra, mật khẩu trong bảng này sẽ được lưu trữ dưới dạng mã Hash.

* + - * 1. Bảng Accounts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | AccountID | Int | >0 | Khóa chính, mã tài khoản |
|  | Username | Vachar | Tối đa 100 kí tự, không rỗng | Tên đăng nhập |
|  | Password | Longtext | Tối thiểu là 6 | Mật khẩu |
|  | SaltMD5 | Longtext | >0 | Mã hash |
|  | PwdResetCode | Longtext | Có thể rỗng | Mã code khi quên mật khẩu |
|  | PwdResetDate | Datetime | Có thể rỗng | Ngày đặt lại mật khẩu |
|  | ReceiveEmail | Varchar | Có thể rỗng | Email nhận code |
|  | RoleID | Int | >0 | Khóa ngoại role |

* + - 1. Roles

Là bảng lưu trữ những vai trò trong hệ thống, đóng vai trò giúp phân định tài khoản nào là admin, giáo viên và sinh viên (người dùng bình thường). Để từ đó, giúp cho việc phân quyền trở nên dễ dàng hơn.

* + - * 1. Bảng Roles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | RoleID | Int | >0 | Khóa chính, mã vai trò |
|  | Name | Vachar | Tối đa 100 kí tự, không rỗng | Tên vai trò |

* + - 1. Coders

Là bảng lưu trữ những thông tin về người dùng, đóng vai trò như là một phần của hệ thống quản lý người dùng. Mỗi bản ghi đại diện cho một người dùng với các thông tin chi tiết ở dưới bảng sau:

* + - * 1. Bảng Coders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | CoderID | Int | >0 | Khóa chính, mã người dùng |
|  | CoderName | Vachar | Tối đa 200 kí tự, không rỗng | Tên người dùng |
|  | CoderEmail | Varchar | Tối đa 200 kí tự, không rỗng | Email người dùng |
|  | PhoneNumber | Varchar | Tối đã 10 kí tự | Số điện thoại |
|  | Avatar | Varchar | Tối đa 500 kí tự, không rỗng | Đường dẫn hình ảnh avatar |
|  | Description | Text | >0 | Giới thiệu bản thân |
|  | CreatedAt | Datetime | Không được rỗng | Ngày tạo |
|  | CreatedBy | Varchar | Tối đa 100 kí tự, không rỗng | Người tạo |
|  | Gender | Int | 3 kí tự (0,1,2) | Giới tính |
|  | UpdatedAt | Datetime | Được rỗng | Ngày cập nhật |
|  | UpdateBy | Longtext | Được rỗng | Người cập nhật |
|  | BirthDay | Datetime | Tối đa 6 | Ngày sinh |

* + - 1. Courses

Là bảng lưu trữ những thông tin liên quan đến khóa học cũng như những mối quan hệ với nó. Quản lý các khóa học trong hệ thống, bao gồm thông tin cơ bản, mô tả, trạng thái, phí, mã giảm giá, hình ảnh, và phân loại.

* + - * 1. Bảng Courses

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | CourseID | Int | >0 | Khóa chính, mã khóa học |
|  | CourseName | Vachar | Tối đa 255 kí tự, không rỗng | Tên khóa học |
|  | Description | Text | Có thể rỗng | Mô tả |
|  | CoderID | Int | >0 | Khóa ngoại coder |
|  | CreatedAt | Datetime | Không được rỗng | Ngày tạo |
|  | UpdatedAt | Datetime | Không được rỗng | Ngày cập nhật |
|  | Status | Int | 2 kí tự (0, 1) | Trạng thái |
|  | CourseCategoryID | Int | >0 | Khóa ngoại category |
|  | Fee | Decimal | 18 kí từ và làm tròn đến số thứ 2 | Giá bán |
|  | OriginalFee | Decimal | 18 kí từ và làm tròn đến số thứ 2 | Giá gốc |
|  | DiscountPercent | Int | >0 | Tỉ lệ giảm giá |
|  | Rating | Double | >=0 và <= 5 | Đánh giá |
|  | IsCombo | Tinyint (1) | 0,1 | Kiểm tra có miễn phí không |
|  | BadgeID | Int | >0 | Khóa ngoại  Badge |
|  | ImageUrl | Longtext | >0 | Đường dẫn hình ảnh |
|  | Overview | Longtext | >0 | Giới thiệu tổng quan |

* + - 1. CourseCategories

Đây là bảng thể loại khóa học giúp phân các khóa học ra nhiều thể loại khác nhau. Phân loại khóa học theo thể loại nhằm hỗ trợ tìm kiếm và tổ chức các khóa học một cách hợp lý.

* + - * 1. Bảng CourseCategories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | CourseCategoryID | Int | >0 | Khóa chính,  Mã thể loại |
|  | Name | Vachar | Tối đa 255 kí tự, không rỗng | Tên thể loại khóa học |
|  | Order | Int | >0 | Vị trí |

* + - 1. Topics

Đây là bảng lưu trữ thông tin về các chủ đề trong hệ thống. Quản lý các chủ đề học tập bên trong khóa học, là đơn vị nhỏ hơn của khóa học giúp chia nhỏ nội dung.

* + - * 1. Bảng Topics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | TopicID | Int | >0 | Khóa chính, mã chủ đề |
|  | CourseID | Int | >0 | Khóa ngoại  Course |
|  | TopicName | Varchar | Tối đa 255 kí tự, không rỗng | Tên chủ đề |
|  | TopicDescription | Longtext | >0 | Mô tả |
|  | CreatedAt | Datetime | Không được rỗng | Ngày tạo |
|  | UpdatedAt | Datetime | Không được rỗng | Ngày cập nhật |
|  | Status | Int | 0,1 | Trạng thái |

* + - 1. Lessons

Bảng lưu trữ thông tin về bài học trong hệ thống. Quản lý bài học trong từng chủ đề, bao gồm tiêu đề, nội dung và trạng thái của bài học.

* + - * 1. Bảng Lessons

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | LessonID | Int | >0 | Khóa chính, mã bài học |
|  | TopicID | Int | >0 | Khóa ngoại  Topic |
|  | LessonTitle | LongText | >0 | Tiêu đề bài học |
|  | LessonContent | Longtext | Được rỗng | Nội dung bài học |
|  | Order | Int | >0 | Vị trí |
|  | CreatedAt | Datetime | Không được rỗng | Ngày tạo |
|  | UpdatedAt | Datetime | Không được rỗng | Ngày cập nhật |
|  | Status | Int | 0,1 | Trạng thái |

* + - 1. LessonProblems

Bảng được sinh ra từ mối quan hệ nhiều – nhiều giữa hai bảng Lessons và Problems. Xác định mối quan hệ nhiều-nhiều giữa các bài học và bài tập (problems) được gắn trong mỗi bài học.

* + - * 1. Bảng LessonProblems

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | ID | Int | >0 | Khóa chính |
|  | LessonID | Int | >0 | Khóa ngoại  Lesson |
|  | ProblemID | Int | >0 | Khóa ngoại  Problem |

* + - 1. Problems

Đây là bảng lưu trữ thông tin bài tập trong hệ thống. Quản lý các bài tập lập trình, bao gồm mô tả, giới hạn thời gian, bộ nhớ, test mẫu và mã mẫu kiểm thử.

* + - * 1. Bảng Problems

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | ProblemID | Int | >0 | Khóa chính, Mã bài tập |
|  | ProblemName | Varchar | Tối đa 255 kí tự, không rỗng | Tên bài tập |
|  | TimeLimit | Int | >0 | Thời gian giới hạn |
|  | MemoryLimit | Int | >0 | Giới hạn bộ nhớ |
|  | ProblemContent | Longtext | >0 | Nội dung bài tập |
|  | ProblemExplanation | Longtext | >0 | Giải thích chi tiết |
|  | TestType | Varchar | Tối đa 50 kí tự | Loại kiểm thử |
|  | TestCode | Longtext | >0 | Code mẫu |
|  | TestProgCompile | Longtext | >0 | Kiểm tra biên dịch |
|  | CoderID | Int | >0 | Khóa ngoại Coder |
|  | Published | Int | 0,1 | Công khai |
|  | TestCompilerID | Int | >0 | Khóa ngoại Compiler |

* + - 1. ProblemCategories

Đây là bảng được sinh ra từ mối quan hệ nhiều – nhiều giữa hai bảng Problems và Categories. Liên kết các bài tập với các thể loại, giúp phân loại bài tập theo nhiều chủ đề hoặc kỹ thuật.

* + - * 1. Bảng ProblemCategories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | ProblemID | Int | >0 | Vừa là khóa chính vừa là khóa ngoại |
|  | CategoryID | Int | >0 | Vừa là khóa chính vừa là khóa ngoại |
|  | Note | Varchar | Tối đa 500 kí tự | Ghi chú |

* + - 1. Categories

Đây là bảng lưu thông tin thể loại bài tập. Danh sách các thể loại bài tập để phục vụ phân nhóm nội dung bài tập cho dễ quản lý và truy xuất.

* + - * 1. Bảng Categories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | CategoryID | Int | >0 | Khóa chính, mã thể loại |
|  | CatName | Varchar | Tối đa 200 kí tự | Tên thể loại |
|  | CatOrder | Int | >0 | Vị trí |

* + - 1. Submissions

Đây là bảng lưu trữ thông tin sự nộp bài. Lưu thông tin chi tiết về bài làm của người dùng bao gồm mã nguồn nộp, kết quả, trạng thái kiểm thử.

* + - * 1. Bảng Submissions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | SubmissionID | Int | >0 | Khóa chính, mã nộp bài |
|  | CoderID | Int | >0 | Khóa ngoại Coder |
|  | CompilerID | Int | >0 | Khóa ngoại Compiler |
|  | ProblemID | Int | >0 | Khóa ngoại Problem |
|  | SubmitTime | Datetime | >0 | Thời gian nộp bài |
|  | SubmissionCode | Longtext | >0 | Mã nguồn khi nộp |
|  | SubmissionStatus | Int | Từ 0 đến 6 | Pending, Accepted, WrongAnswer, TimeLimitExceeded, MemoryLimitExceeded, RuntimeError, CompilationError |
|  | TestResult | Longtext | >0 | Kết quả kiểm thử |
|  | MaxMemorySize | float | >0 | Bộ nhớ tối đa |
|  | MaxTimeDuration | Int | >0 | Thời gian tối đa |
|  | CreatedAt | Datetime | >0 | Ngày tạo |
|  | UpdatedAt | Datetime | >0 | Ngày cập nhật |

* + - 1. Compilers

Đây là bảng lưu trữ thông tin của biên dịch. Lưu trữ thông tin các trình biên dịch được hỗ trợ trong hệ thống như tên, đường dẫn, tuỳ chọn biên dịch.

* + - * 1. Bảng Compilers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | CompilerID | Int | >0 | Khóa chính, mã biên dịch |
|  | CompilerName | Varchar | Tối đa 255 kí tự | Tên trình biên dịch |
|  | CompilerPath | Varchar | Tối đa 500 kí tự | Đường dẫn biên dịch |
|  | CompilerOption | Int | >0 | Lựa chọn biên dịch |
|  | CompilerExtension | Varchar | Tối đa 50 kí tự | Đuôi file biên dịch |

* + - 1. TestCases

Đây là bảng lưu trữ thông tin về các bộ kiểm thử cho bài toán. Quản lý các bộ test (đầu vào, đầu ra) dùng để kiểm thử bài tập lập trình của người dùng.

* + - * 1. Bảng TestCases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | TestCaseID | Int | >0 | Khóa chính, mã bộ kiểm thử |
|  | ProblemID | Int | >0 | Khóa ngoại Problem |
|  | TestCaseOrder | Int | >0 | Vị trí bộ kiểm thử |
|  | SampleTest | Int | >0 | Test ví dụ |
|  | PreTest | Int | >0 | Test ẩn |
|  | Input | Longtext | >0 | Đầu vào |
|  | Output | Longtext | >0 | Đầu ra |

* + - 1. TestRuns

Đây là bảng lưu trữ thông tin kết quả chạy thử của một bài nộp trên từng testcase. Lưu kết quả chi tiết khi một submission được chạy qua từng test case, bao gồm thời gian, bộ nhớ, và log.

* + - * 1. Bảng TestRuns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | TestRunID | Int | >0 | Khóa chính, mã bộ kiểm thử |
|  | SubmissionID | Int | >0 | Khóa ngoại Submission |
|  | TestCaseID | Int | >0 | Khóa ngoại TestCase |
|  | TimeDuration | Int | >0 | Thời gian chạy trương trình (ms) |
|  | MemorySize | Int | >0 | Bộ nhớ sử dụng (KB) |
|  | TestOutput | Longtext | >0 | Kết quả chương trình in ra |
|  | Result | Longtext | >0 | Kết quả mong đợi |
|  | CheckerLog | Longtext | >0 | Ghi chú chi tiết từ bộ kiểm tra kết quả |

* + - 1. Solved

Đây là bảng lưu trữ thông tin đánh dấu người dùng đã làm bài tập này. Ghi nhận bài tập nào đã được người dùng giải thành công, giúp theo dõi tiến độ học tập.

* + - * 1. Bảng Solved

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | CoderID | Int | >0 | Vừa là khóa chính vừa là khóa ngoại |
|  | ProblemID | Int | >0 | Vừa là khóa chính vừa là khóa ngoại |

* + - 1. Matches

Đây là bảng lưu trữ thông tin điểm số của mỗi người dùng. Lưu điểm số người dùng đạt được cho từng bài tập trong các bài học, hỗ trợ thống kê và đánh giá.

* + - * 1. Bảng Matches

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | MatchID | Int | >0 | Khóa chính, mã điểm |
|  | CoderID | Int | >0 | Khóa ngoại Coder |
|  | LessonProblemID | Int | >0 | Khóa ngoại LessonProblem |
|  | Point | Int | >0 | Điểm |

* + - 1. Enrollments

Đây là bảng lưu trữ thông tin đánh dấu người dùng nào đã đăng ký khóa học. Ghi nhận thông tin người dùng đã đăng ký học các khóa học nào, làm cơ sở cho quản lý học viên.

* + - * 1. Bảng Enrollments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | EnrollmentID | Int | >0 | Khóa chính, mã đăng ký |
|  | CourseID | Int | >0 | Khóa ngoại Course |
|  | CoderID | Int | >0 | Khóa ngoại Coder |
|  | EnrolledAt | Datetime | Không được rỗng | Ngày đăng ký khóa học |

* + - 1. Badges

Bảng lưu trữ thông tin về các thẻ tag. Lưu thông tin các huy hiệu/tags dùng để đánh dấu đặc điểm nổi bật của khóa học, giúp tăng nhận diện.

* + - * 1. Bảng Badges

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | BadgeID | Int | >0 | Khóa chính, mã thẻ tag |
|  | Name | Varchar | Tối đa 50 kí tự | Tên thẻ tag |
|  | Description | Varchar | Tối đa 255 kí tự | Mô tả |
|  | Color | Varchar | Tối đa 7 kí tự | Mã màu |

* + - 1. Blogs

Bảng lưu trữ thông tin về các bài viết. Quản lý bài viết (blog) do người dùng đăng, bao gồm nội dung, ngày đăng, trạng thái công khai và lượt xem.

* + - * 1. Bảng Blogs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | BlogID | Int | >0 | Khóa chính, mã bài viết |
|  | Title | Varchar | Tối đa 200 kí tự | Tiêu đề bài viết |
|  | BlogDate | DateTime | Không được rỗng | Ngày đăng bài viết |
|  | Content | Longtext | >0 | Nội dung bài viết |
|  | Published | Int | 0,1 | Trạng thái công khai |
|  | PinHome | Int | 0,1 | Hiển thị ở trang Home |
|  | CoderID | Int | >0 | Khóa ngoại Coder |
|  | BlogImage | Longtext | >0 | Ảnh đại diện cho bài viết |
|  | ViewCount | Int | >0 | Lượt xem của bài viết |

* + - 1. Reviews

Bảng lưu trữ thông tin về đánh giá khóa học. Lưu đánh giá của người học về khóa học, bao gồm số sao, nội dung nhận xét, và thời gian tạo đánh giá.

* + - * 1. Bảng Reviews

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | ReviewID | Int | >0 | Khóa chính, mã đánh giá |
|  | CourseID | Int | >0 | Khóa ngoại Course |
|  | CoderID | Int | >0 | Khóa ngoại Coder |
|  | Rating | Int | >0 và <5 | Đánh giá của người dùng |
|  | Content | Varchar | Tối đa 1000 kí tự | Nội dung đánh giá |
|  | CreatedAt | Datetime | Không được rỗng | Ngày tạo |

* + - 1. Comments

Bảng lưu trữ thông tin những bình luận của người dùng. Quản lý các bình luận của người dùng trên blog hoặc khóa học, hỗ trợ phản hồi và thảo luận.

* + - * 1. Bảng Comments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | CommentID | Int | >0 | Khóa chính, mã bình luận |
|  | CoderID | Int | >0 | Khóa ngoại Coder |
|  | Content | Longtext | >0 | Nội dung bình luận |
|  | CommentTime | Datetime | Không được rỗng | Thời gian bình luận |
|  | BlogID | Int | >0 | Khóa ngoại Blog |
|  | CourseID | Int | >0 | Khóa ngoại Course |
|  | ParentCommentID | Int | >0 | Khóa ngoại Comment |

* + - 1. Progresses

Bảng lưu trữ thông tin về tiến độ hoàn thành khóa học. Theo dõi tiến độ học tập của người dùng với từng khóa học, bài học hoặc bài tập bằng phần trăm hoàn thành hoặc đánh dầu hoàn thành.

* + - * 1. Bảng Progresses

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ghi chú** |
|  | ProgressID | Int | >0 | Khóa chính, mã tiến độ |
|  | CoderID | Int | >0 | Khóa ngoại Coder |
|  | ObjectID | Int | >0 | ID của đối tượng (lesson, topic, course) |
|  | ObjectType | Int | >0 | Kiểu đối tượng |
|  | Percent | Double | >0 | Phần trăm hoàn thành |
|  | LastUpdated | Datetime | Không được rỗng | Thời gian cập nhật lần cuối |

* + 1. Thiết kế xây dựng API

Hệ thống được nâng cấp và làm mới dựa trên nền tảng hệ thống cũ là NTUCoder sử dụng RESTful API để xây dựng ứng dụng. Điều này nhằm mục đích là đảm bảo tính linh hoạt, dễ mở rộng và dễ tích hợp. Vì là API nên là dễ dàng giao tiếp giữa hai giao diện client (FE) và server (BE) giúp người dùng có thể tương tác dễ dàng các chức năng như là đăng nhập, nộp bài tập hay là đăng bài viết, …

Mục đích của việc thiết kế API là:

* **Tái sử dụng và dễ mở rộng**: các chức năng được thể hiện dưới dạng endpoint cho nên cho phép người dùng có thể dễ dàng ứng dụng trên mọi nền tảng khác nhau (mobile, web, admin tool).
* **Đảm bảo bảo mật dữ liệu**: áp dụng cơ chế xác thực bằng token (JWT).
* **Tách biệt hệ thống**: giao diện người dùng và giao diện xử lý được tách biệt hoàn toàn, giúp cho người lập trình có thể dễ dàng bảo trì và nâng cấp hệ thống.
  + - 1. API cho lớp Authorization

Nhóm API này phụ trách chức năng xác thực và phân quyền người dùng. Toàn bộ hoạt động đăng nhập, đăng xuất, đăng nhập với Google, và khôi phục mật khẩu đều được quản lý tập trung tại đây.

* **POST /api/Auth/login**: Đăng nhập với username và password, sau đó tạo ra 1 token để xác thực tài khoản.
* **POST /api/Auth/logout**: Đăng xuất và xóa token.
* **GET /api/Auth/me**: Trả về thông tin của coder đang đăng nhập, dựa trên token xác thực JWT.
* **POST /api/Auth/google-login:** Đăng nhập bằng tài khoản Google.
* **POST /api/Auth/send-reset-code:** Gửi mã xác thực để đặt lại mật khẩu.
* **POST /api/Auth/verify-reset-code:** Xác minh mã đặt lại mật khẩu.
* **POST /api/Auth/reset-password:** Đặt lại mật khẩu mới.
  + - 1. API cho lớp CodeExecute

Đây là nhóm API đặc thù trong hệ thống chấm bài. Nó đảm nhận việc biên dịch và chạy mã nguồn của người dùng thông qua môi trường Docker cô lập.

* **POST /api/CodeExecute/{submissionId}:** Chạy code từ submission có sẵn.
* **POST /api/CodeExecute/try-run:** Thử chạy code với test case mẫu.
* **POST /api/CodeExecute/multi-sub:** Chạy nhiều submission cùng lúc.
  + - 1. API cho lớp Coder

API cho lớp Coder cung cấp các chức năng quản lý người học lập trình (coder), bao gồm tra cứu, thống kê và quản lý thông tin.

* **GET /api/Coder:** Danh sách coder (hỗ trợ lọc, phân trang và sắp xếp).
* **GET /api/Coder/search?keyword=&page=&pageSize=:** Tìm kiếm coder theo từ khóa, phân trang.
* **GET /api/Coder/{id}:** Lấy chi tiết coder theo ID.
* **GET /api/Coder/top-hightest:** Lấy 3 coder có điểm cao nhất.
* **GET /api/Coder/list-ranking?q=&page=&pageSize=:** Lấy danh sách coder theo bảng xếp hạng, có thể tìm kiếm theo tên (q).
* **POST /api/Coder:** Tạo coder mới.
* **PUT /api/Coder/{id}:** Cập nhật thông tin coder.
* **DELETE /api/Coder/{id}:** Xóa coder theo ID.
  + - 1. API cho lớp Compiler

API cho lớp Coder cung cấp các chức năng quản lý người học lập trình (coder), bao gồm tra cứu, thống kê và quản lý thông tin.

* **GET /api/Compiler:** Danh sách ngôn ngữ lập trình hỗ trợ.
* **GET /api/Compiler/{id}:** Lấy chi tiết ngôn ngữ lập trình theo ID.
* **POST /api/Compiler:** Thêm ngôn ngữ lập trình mới.
* **PUT /api/Compiler/{id}:** Cập nhật thông tin ngôn ngữ lập trình.
* **DELETE /api/Compiler/{id}:** Xóa ngôn ngữ lập trình.
  + - 1. API cho lớp Course

Quản lý các khóa học trên hệ thống LMS (Learning Management System).

* **GET /api/Course**: Danh sách khóa học (có hỗ trợ lọc, phân trang và sắp xếp).
* **GET /api/Course/search?keyword=:** Tìm kiếm khóa học theo từ khóa.
* **GET /api/Course/{id}:** Chi tiết khóa học.
* **GET /api/Course/problem-count?courseId=:** Đếm số bài toán trong khóa học.
* **POST /api/Course:** Tạo khóa học mới (yêu cầu xác thực).
* **PUT /api/Course/{id}:** Cập nhật khóa học (yêu cầu xác thực).
* **DELETE /api/Course/{id}:** Xóa khóa học.
  + - 1. API cho lớp Problem

Quản lý các bài tập lập trình (problem), là trung tâm của toàn bộ hệ thống.

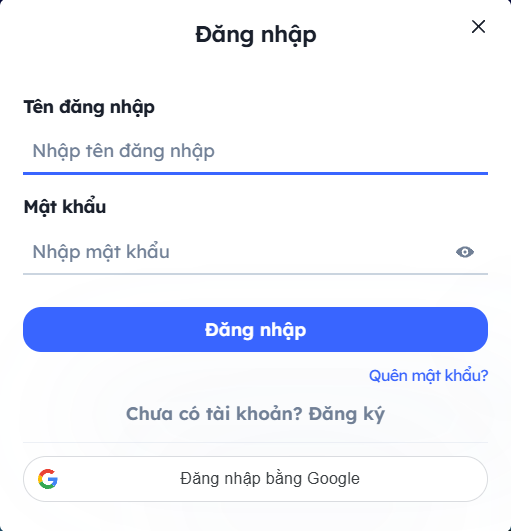
* **GET /api/Problem:** Lấy danh sách bài tập (hỗ trợ phân trang, sắp xếp, lọc theo trạng thái published).
* **GET /api/Problem/{id}:** Lấy chi tiết bài tập theo ID.
* **POST /api/Problem**: Tạo bài tập mới (yêu cầu xác thực).
* **PUT /api/Problem/{id}:** Cập nhật bài tập theo ID (yêu cầu xác thực).
* **DELETE /api/Problem/{id}:** Xóa bài tập theo ID.
  + - 1. API cho lớp Submission

Quản lý quá trình nộp bài và lưu trữ kết quả chấm bài.

* **GET /api/Submission/all?sortField=&ascending=&compilerFilter=:** Lấy danh sách tất cả bài nộp (hỗ trợ tìm kiếm, lọc và phân trang).
* **POST /api/Submission/submit:** Nộp bài làm và thực thi kiểm thử (yêu cầu xác thực).
* **GET** **/api/Submission/{id}:** Lấy thông tin bài nộp theo ID.
* **PUT /api/ Submission /{id}:** Cập nhật bài nộp.
* **DELETE /api/Submission/{id}:** Xóa bài nộp theo ID.
* **GET /api/Submission/history?problemId=&sortField=&ascending=&coderId=:** Lấy lịch sử nộp bài theo mã bài tập hoặc là theo mã coder.
  + - 1. API cho lớp TestCase

Quản lý các test case dùng để kiểm thử bài nộp.

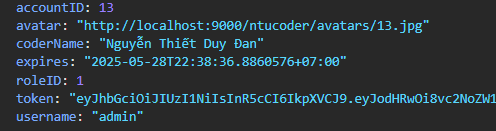
* **GET /api/TestCase?problemID=&sortField=&ascending=:** Lấy danh sách test case của một bài tập theo problemID (hỗ trợ phân trang và sắp xếp).
* **POST /api/TestCase:** Tạo mới một test case cho bài tập (yêu cầu body chứa TestCaseDTO).
* **GET /api/TestCase/{id}:** Lấy thông tin chi tiết của test case theo ID.
* **PUT /api/TestCase/{id}:** Cập nhật test case theo.
* **DELETE /api/TestCase/{id}:** Xóa test case theo ID.
* **GET /api/TestCase/count?problemId=:** Lấy tổng số test case của một bài tập.
* **GET /api/TestCase/sampleTest?problemId=:** Lấy test case mẫu (sample test) của một bài tập.
  + 1. Thiết kế giao diện kết nối API
       1. Giao diện đăng nhập

Giao diện đăng nhập

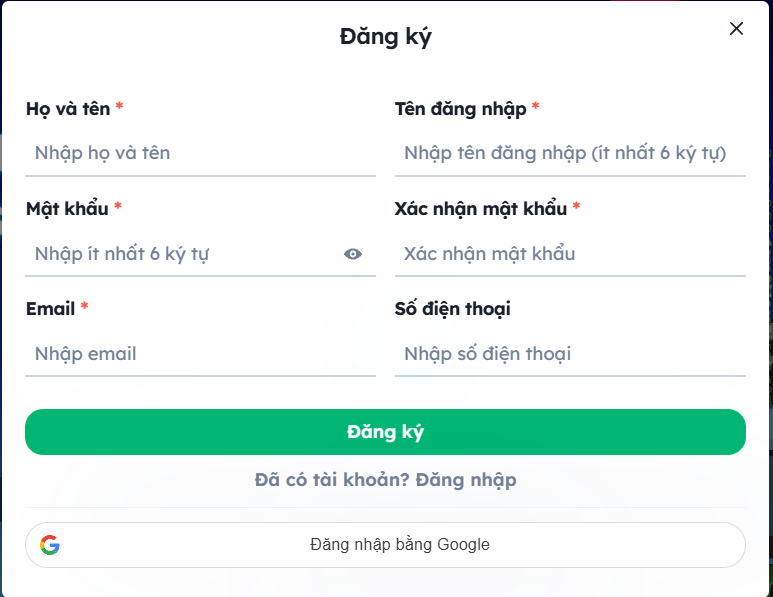
Giao diện này được thể hiện dưới dạng modal khi bấm vào nút đăng nhập sẽ xuất hiện. Nó bao gồm hai input là tên đăng nhập và mật khẩu. Mật khẩu có thể ẩn hoặc hiển thị và còn có chỗ để bấm quên mật khẩu. Ngoài ra, còn có thể đăng nhập bằng tài khoản Google. Khi chưa đăng ký tài khoản thì ta cũng có thể bấm vào “Chưa có tài khoản? Đăng ký” thì ta sẽ chuyển đến phần đăng ký. Sau khi bấm đăng nhập thì người dùng sẽ được chuyển đến trang chủ chính và có thông báo (toast) đăng nhập thành công.

Ở giao diện này thì ta dùng những API sau:

* **POST /api/Auth/login**



Kết quả trả về của API trên

* Sau đó thì ta được **GET /api/Auth/me** dùng để xác thực và lấy thông tin người dùng.
  + - 1. Giao diện đăng ký

Giao diện đăng ký

Giao diện này là một modal gồm những ô input như là: họ và tên, tên đăng nhập, mật khẩu, xác nhận mật khẩu, … Ở mỗi ô input sẽ có một dấu sao đỏ để báo cho người dùng đây là ô bắt buộc nhập. Khi mà người dùng không nhập gì sẽ thông báo lỗi như sau:

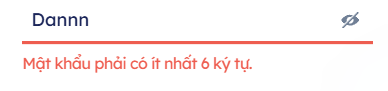


Giao diện lỗi khi bỏ trống

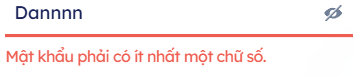
Ngoài ra còn có một số lỗi liên quan đến mật khẩu như sau:



Lỗi thiếu chữ in hoa



Lỗi thiếu kí tự



Lỗi thiếu chữ số

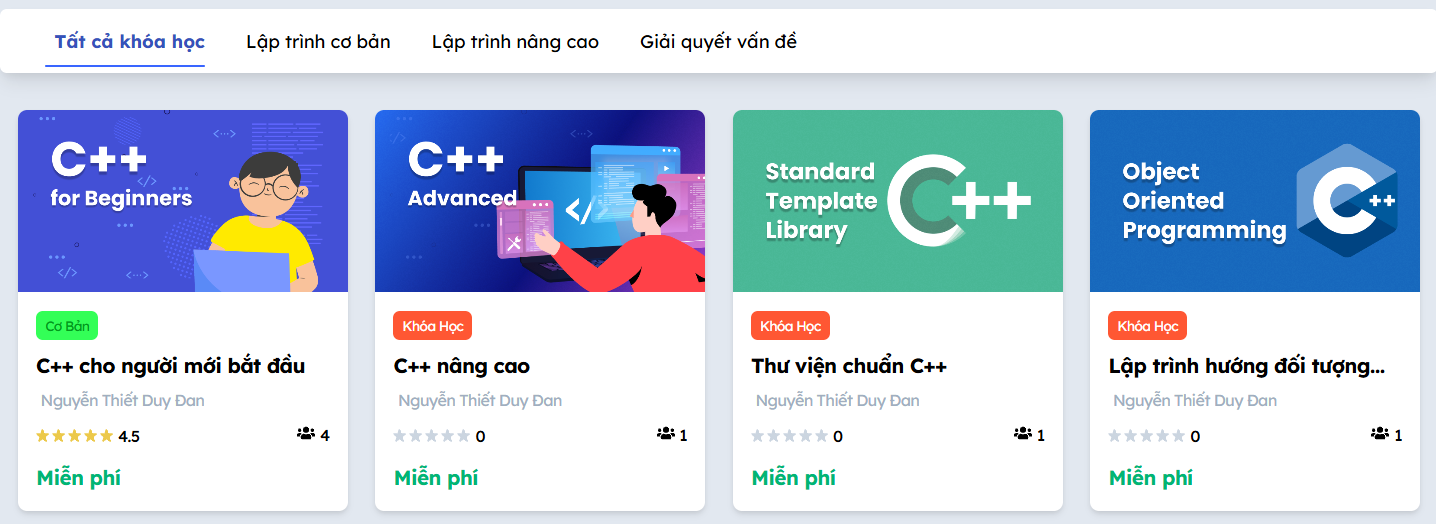


Lỗi thiếu ký tự đặc biệt

Lý do để những kiểu bắt buộc này là để tạo ra những mật khẩu thật an toàn và bảo mật. Ngoài ra còn một số lỗi khác ở những input khác.

Ở giao diện này ta dùng API sau **POST /api/Coder,** lý do dùng API này là vì khi đăng ký người dùng mới sẽ tương ứng với tạo mới người dùng. Tuy nhiên, có một điều khác ở đây là khi đăng ký ở đây thì chỉ có vai trò là người dùng – tức là role = 2.

* + - 1. Giao diện học tập (các khóa học)

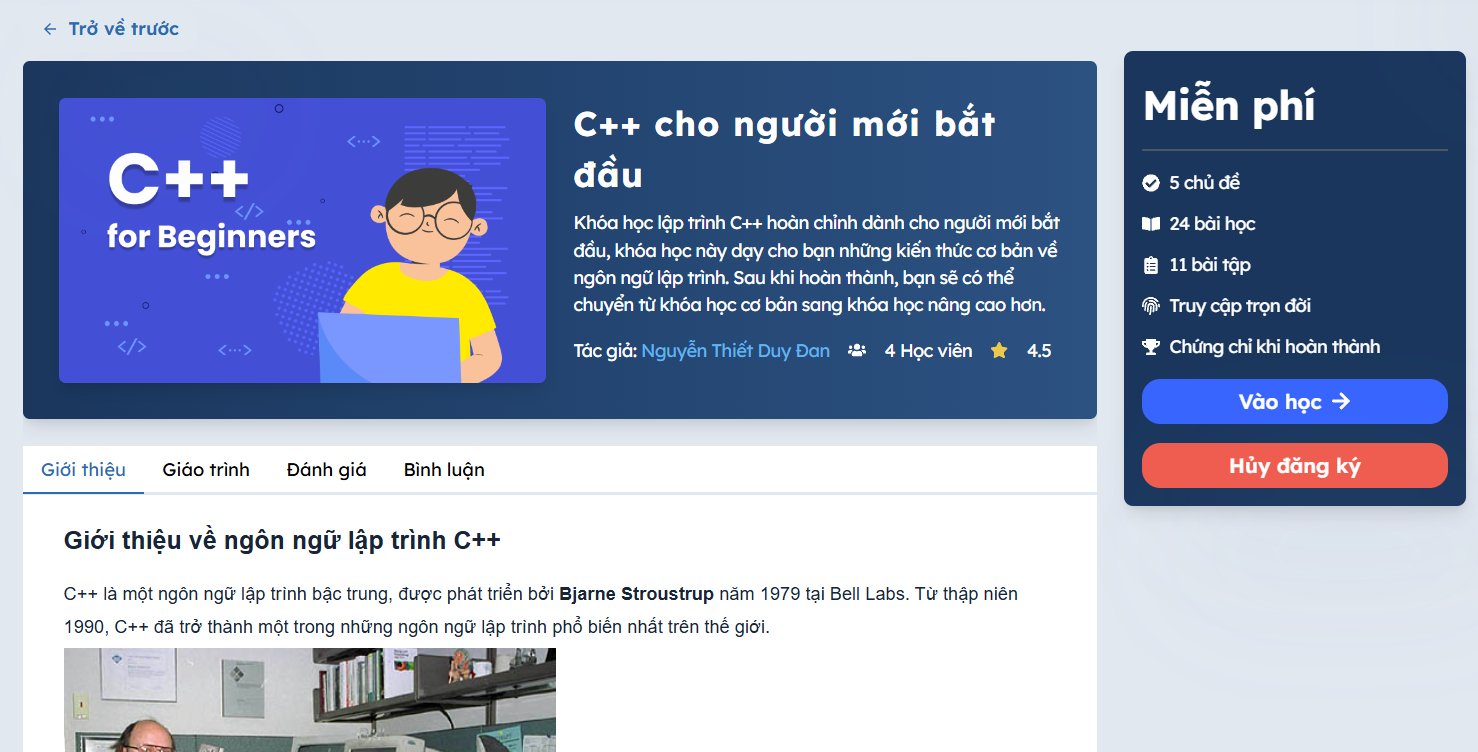


Giao diện danh sách các khóa học

Ở giao diện này sẽ cho người dùng xem danh sách các khóa học đã được công khai, ở đây cũng phân chia ta nhiều loại khóa học để giúp người dùng có thể tìm kiếm khóa học theo ý muốn. Ở mỗi khóa học tương ứng với một card nó bao gồm những phần như sau: Ảnh đại diện, thẻ tag, tên khóa học, người tạo, độ đánh giá, số người đăng ký và dưới dùng là hiển thị loại khóa học – khóa học miễn phí hoặc có tiền (nếu có thêm tiền).

Hiện tại, danh sách khóa học này được gọi thông qua API như sau: “[**https://localhost:7015/api/Course?Page=1&PageSize=10&ascending=true**](https://localhost:7015/api/Course?Page=1&PageSize=10&ascending=true)**”,** có thể thêm phân trang ở dưới bởi vì trong API có chức năng phân trang, nhưng hiện tại theo API thì nó đang là lấy 10 khóa học đầu tiên. Danh sách này có thể hiển thị kể cả người dùng chưa đăng nhập hoặc là ở bất kỳ vai trò gì.

* + - 1. Giao diện chi tiết khóa học



Giao diện ở chi tiết khóa học (bên trái)

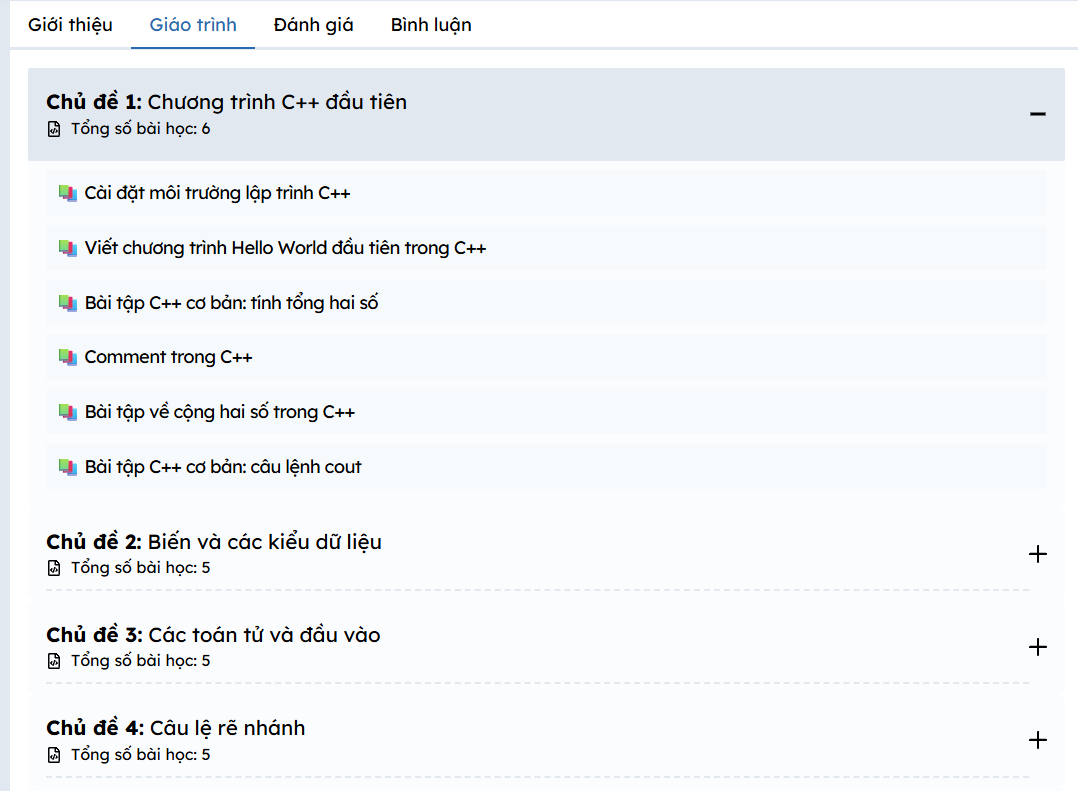
Giao diện này sẽ cho ta thấy toàn bộ những thông tin liên quan đến khóa học được bấm vào. Với phần đầu tiên trên cùng là các thông tin đã được xem ở danh sách và bổ sung thêm là phần mô tả ngắn gọn về khóa học đó. Bên dưới sẽ là những tab liên quan đến nội dung của khóa học bao gồm:

* **Giới thiệu:** Đây là phần giới thiệu tổng quát về khóa học cùng những một số khái niệm liên quan đến khóa học đang được quan tâm. Ở tab này website đã gọi API **GET** [**https://localhost:7015/api/Course/13**](https://localhost:7015/api/Course/13)dùng để lấy thông tin khóa học với CourseID là 13.



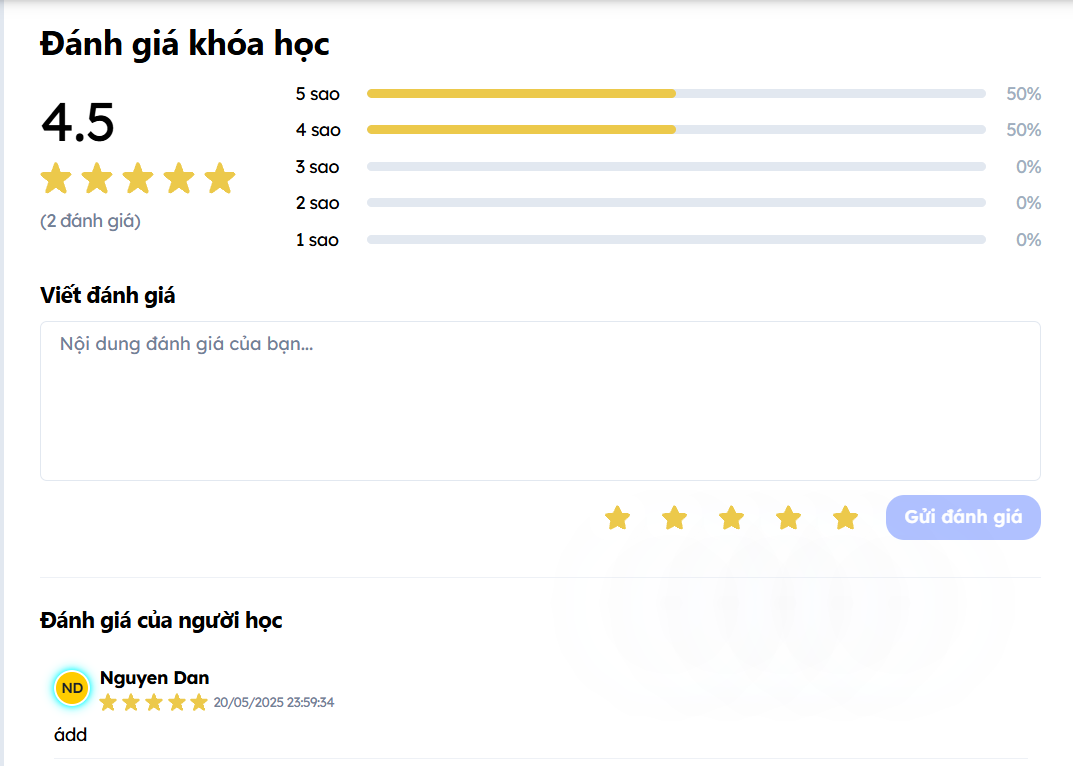
Tab giới thiệu về khóa học

* **Giáo trình:** Là phần hiển thị danh sách các chủ đề và các bài học liên quan đến khóa học. Ở tab này danh sách topic được lấy từ API chi tiết khóa học trên rồi từ đó ta có các TopicID và đưa nó vào mảng rồi lấy vòng lặp để gọi lần lượt các chi tiết Topic trong đó sẽ có thông tin về danh sách của Lesson của mỗi Topic, API đã được sử dụng **GET https://localhost:7015/api/Topic.**



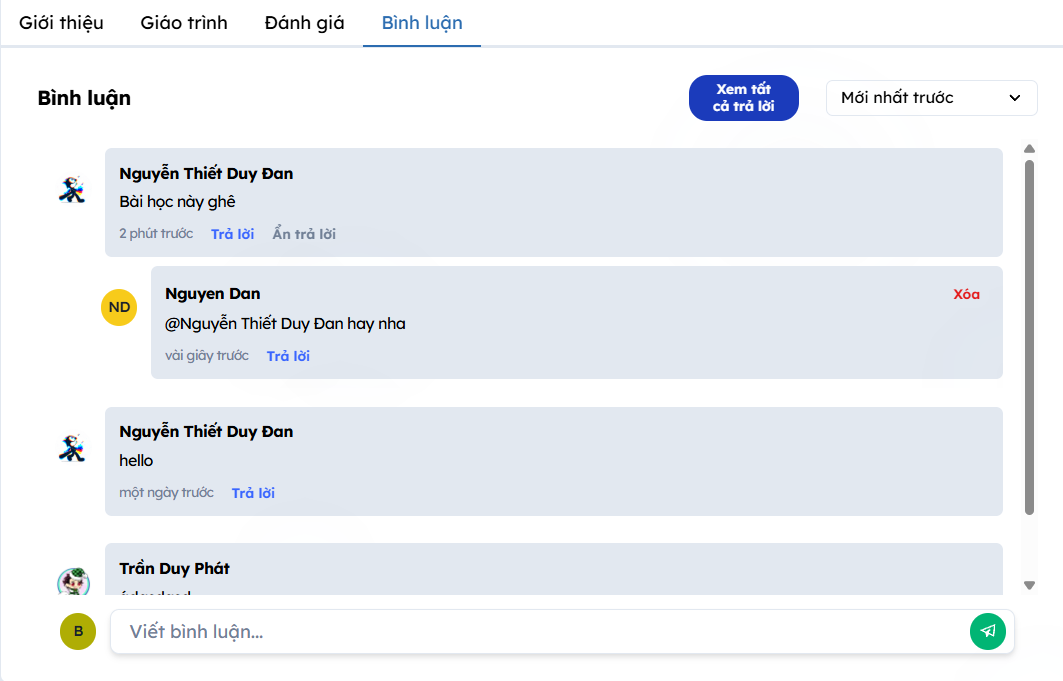
Tab danh sách chủ đề và bài học

* **Đánh giá:** Đây là phần dành cho người dùng sau khi trải nghiệm khóa học muốn đánh giá hoặc là muốn xem những đánh giá những người học khác. Ở trong tab này người dùng cần phải **đăng nhập** mới có thể đăng đánh giá của mình được. Tab này sử dụng API **GET** [**https://localhost:7015/api/Review?courseId=13**](https://localhost:7015/api/Review?courseId=13)để lấy các thông tin liên quan đến đánh giá với courseId bằng 13.

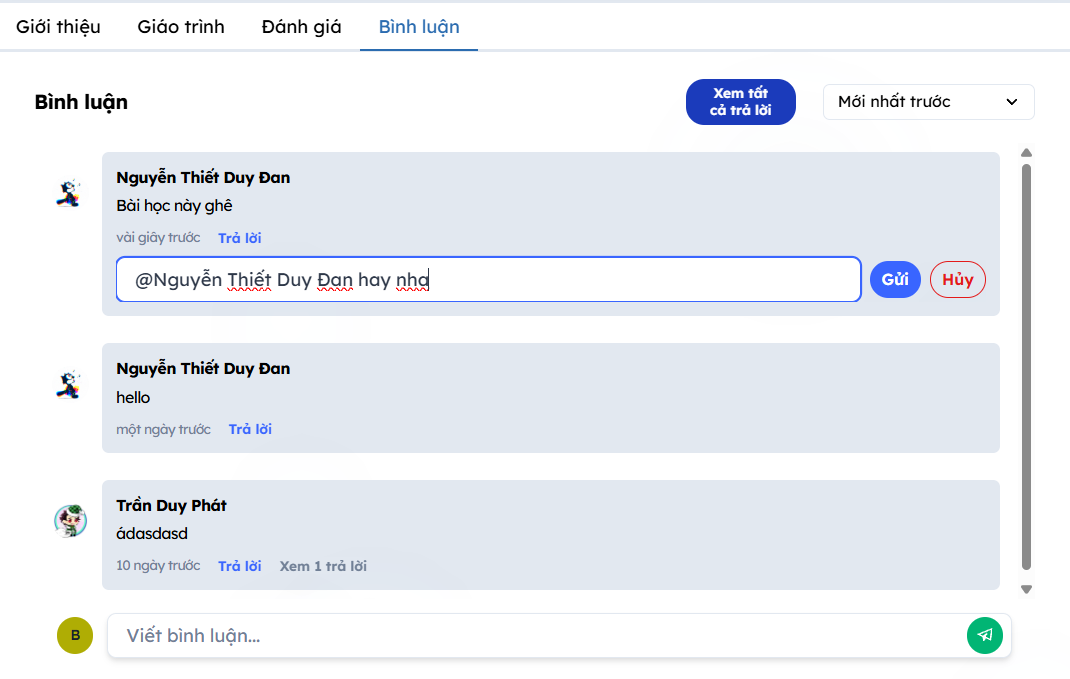


Tab đánh giá của người dùng

* **Bình luận:** Nơi đây là phần dành cho người dùng bình luận với giao để có thể trao đổi kiến thức đã học hoặc là học hỏi từ những người trước. Người dùng cũng có thể trả lời bình luận của người khác. Tuy nhiên, người dùng cần phải đăng nhập mới có thể xem cũng như là bình luận và tab này sử dụng API như sau: GET [**https://localhost:7015/api/comments?courseId=13&page=1&pageSize=5&sortBy=CommentTime&ascending=false**](https://localhost:7015/api/comments?courseId=13&page=1&pageSize=5&sortBy=CommentTime&ascending=false)với các thông số như sau:
* courseId: mã của khóa học hiện tại
* page: trang hiện tại
* pageSize: số lượng có thể hiển thị
* sortBy=CommentTime: đang sắp xếp theo trường CommentTime
* ascending=false: sắp xếp theo mới nhất

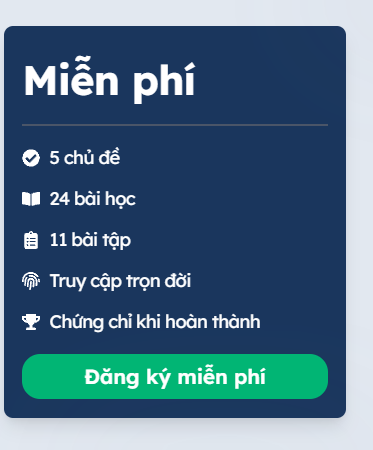


Tab các bình luận của người dùng



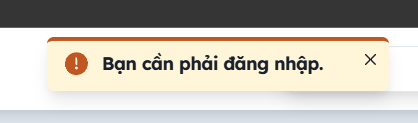
Người dùng trả lời bình luận của người khác

Phía bên phải của giao diện của chi tiết khóa học cũng có một ô để thể hiện các thông tin liên quan để khóa học, chẳng hạn như là số lượng chủ đề, số lượng bài học, số lượng bài tập và nút đăng ký khóa học.



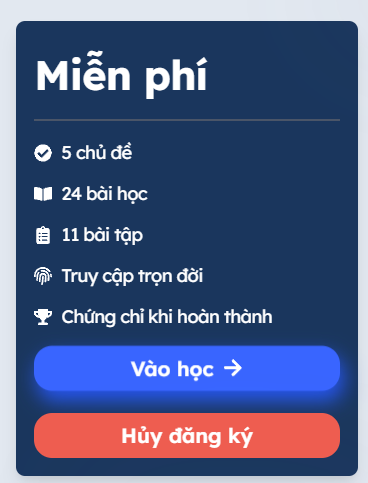
Giao diện ở chi tiết khóa học (bên phải)

Riêng nút đăng ký khóa học này cần phải đăng nhập thì mới có thể bấm được nếu không thì nó sẽ thông báo như hình bên dưới:



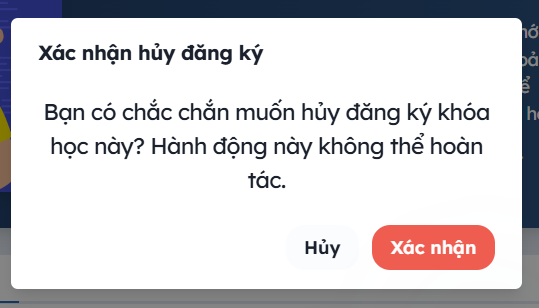
Cảnh báo cần đăng nhập

Sau khi đăng nhập xong, khi người dùng bấm đăng ký cũng xuất hiện thông báo tương tự nhưng sẽ là thông báo thành công và nút đăng ký biến đổi thành hai nút khác. Để có thể đăng ký thì ta gọi đến API như sau **POST** [**https://localhost:7015/api/Enrollment**](https://localhost:7015/api/Enrollment)để tạo mới một lượt đăng ký và mỗi khi vào trang chi tiết thông tin khóa học thì đều gọi tới **GET** [**https://localhost:7015/api/Enrollment/CheckEnrollment?courseId=13&coderID=13**](https://localhost:7015/api/Enrollment/CheckEnrollment?courseId=13&coderID=13)để kiểm tra xem người dùng này đã đăng ký khóa học này chưa.



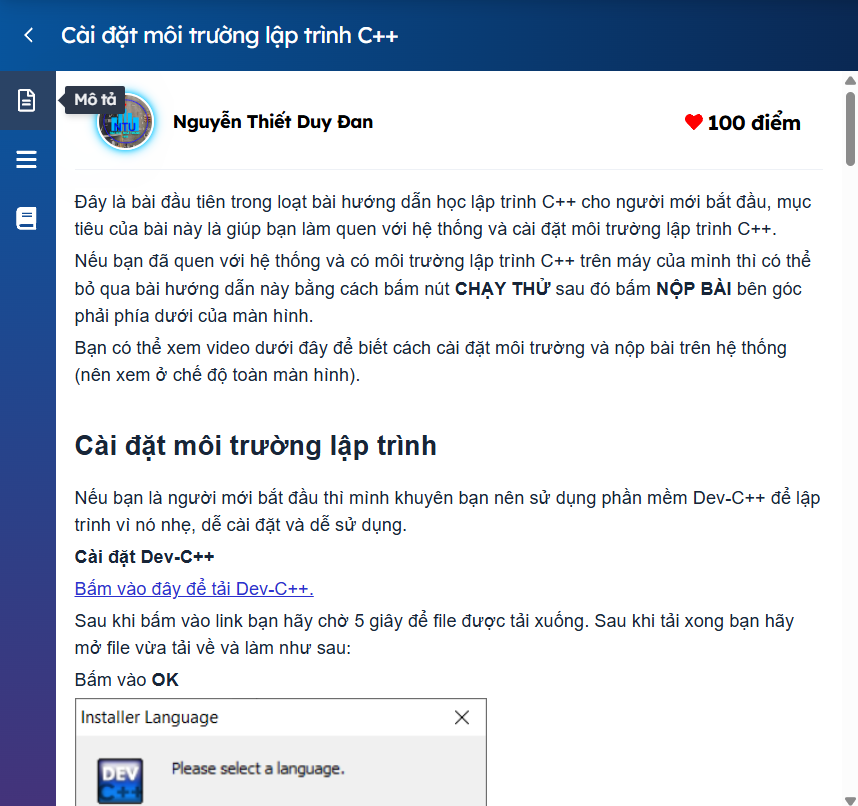
Sau khi bấm đăng ký

Khi người dùng muốn hủy đăng ký, web sẽ hiển thị một modal để hỏi người dùng có chắc chắn có muốn hủy hay không để tránh cho người dùng bấm nhầm rồi gây ra mất những tích lũy đã có được thông qua những việc hoàn thành các bài tập.



Giao diện xác nhận hủy đăng ký khóa học

* + - 1. Giao diện khi làm bài tập



Chi tiết bài học – phần mô tả

Ở phần này sẽ cho người dùng xem những khái niệm liên quan đến bài học và điểm số của bài học đó khi làm xong. Tác dụng của phần này là giúp người học có thể vừa học, vừa ôn lại kiến thức và vừa lập trình.



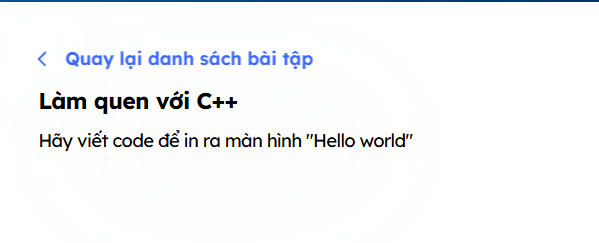
Chi tiết bài học – phần giáo trình

Phần này cũng giống phần giáo trình ở ngoài chi tiết khóa học, chỉ khác ở một vài chỗ là sẽ có thông tin số bài tập của mỗi bài học và khi người dùng hoàn thành số bài tập đó sẽ đánh dấu tích. Sau khi bấm vào bài học sẽ hiển thị ra danh sách các bài tập.

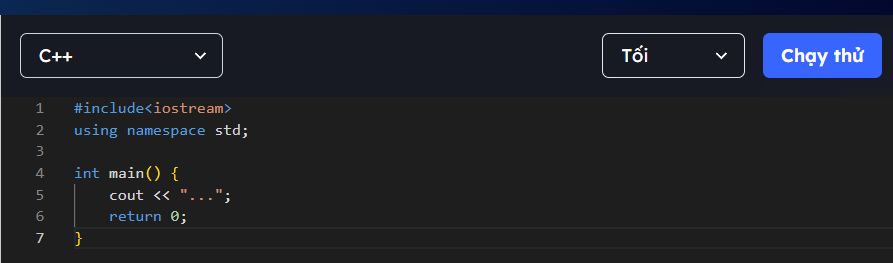


Danh sách các bài tập

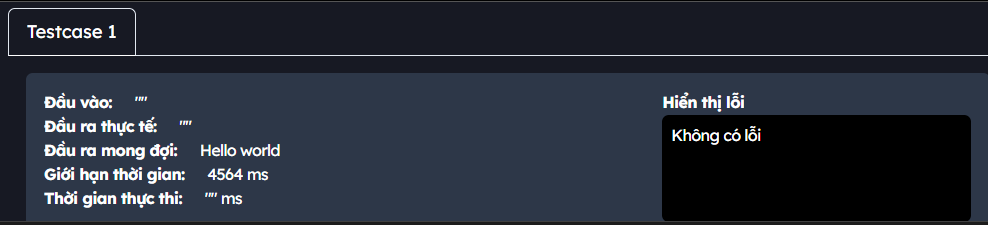
Khi bấm vào một bài tập sẽ ra trang chi tiết bài học. Đồng thời bên tay phải cũng load dữ liệu cho code mẫu và testcase với đầu ra mong muốn.



Chi tiết bài tập

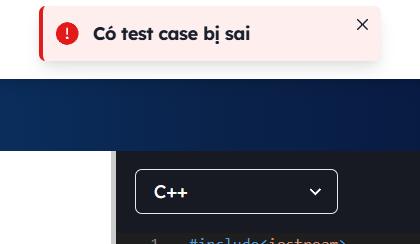


Code mẫu



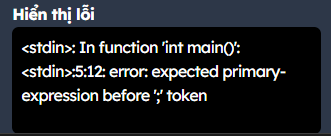
Testcase của bài tập

Trước khi nộp bài phải bấm chạy thử thì nút nộp bài mới được hoạt động. Dưới đây là hình ảnh sau khi bấm nộp bài với dữ liệu sai.



Thông báo bị sai testcase

Khi mà có lỗi liên quan đến cấu trúc code thì nó sẽ xuất hiện ở ô hiển thị lỗi.

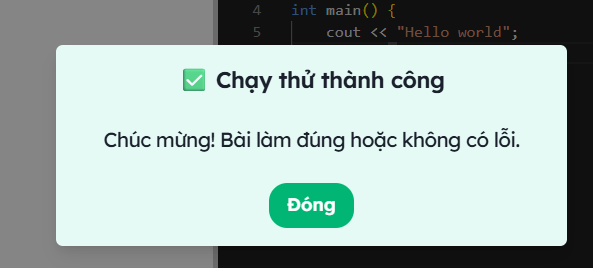


Hiển thị lỗi cấu trúc code



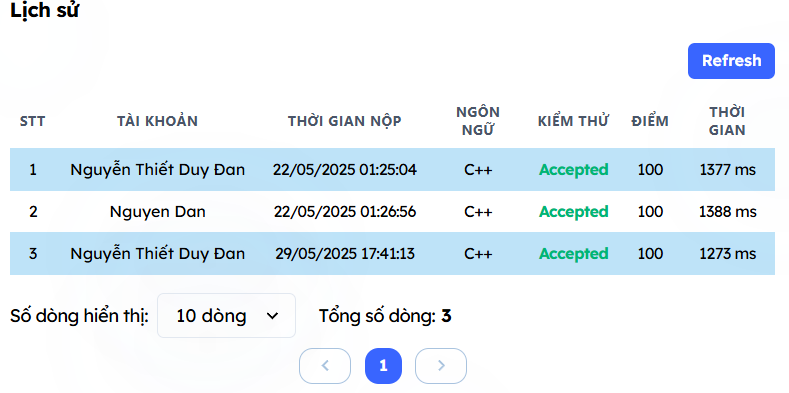
Khi chạy thử thành công

Sau khi chạy thử thành công thì nút nộp bài sẽ được cho phép bấm. Và sau khi nộp bài thì ta sẽ xuất hiện một ô thông báo.



Thông báo nộp bài thành công.

Lúc này bên tay trái sẽ có phần hiển thị lịch sử nộp bài của cá nhân cũng như của những người khác

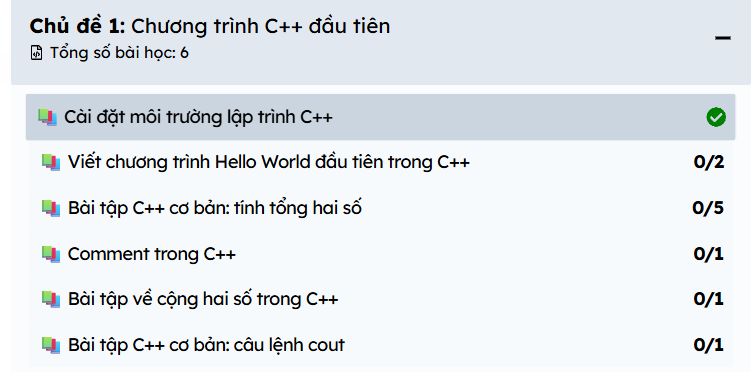


Lịch sử nộp bài

Ngoài ra ở danh sách bài tập cũng đánh dấu tích khi bài tập đó nộp thành công và ở ngoài danh sách bài học cũng đánh dấu bài đó hoàn thành nếu bài đó có 1 bài tập ở trong còn nếu có nhiều bài tập thì nó sẽ đánh **số bài tập / tổng số bài tập**.

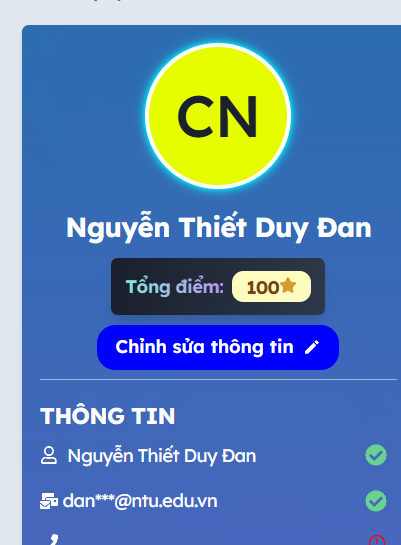


Đánh dấu hoàn thành bài tập



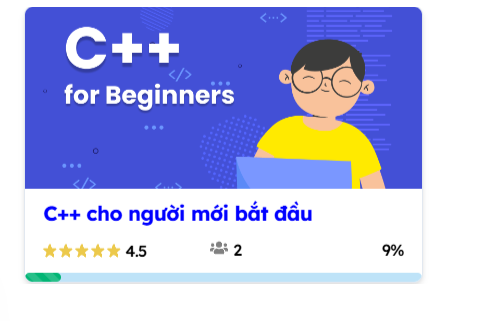
Đánh dấu bài học đó hoàn thành

Và trong trang cá nhân người đó cũng được 100 điểm.



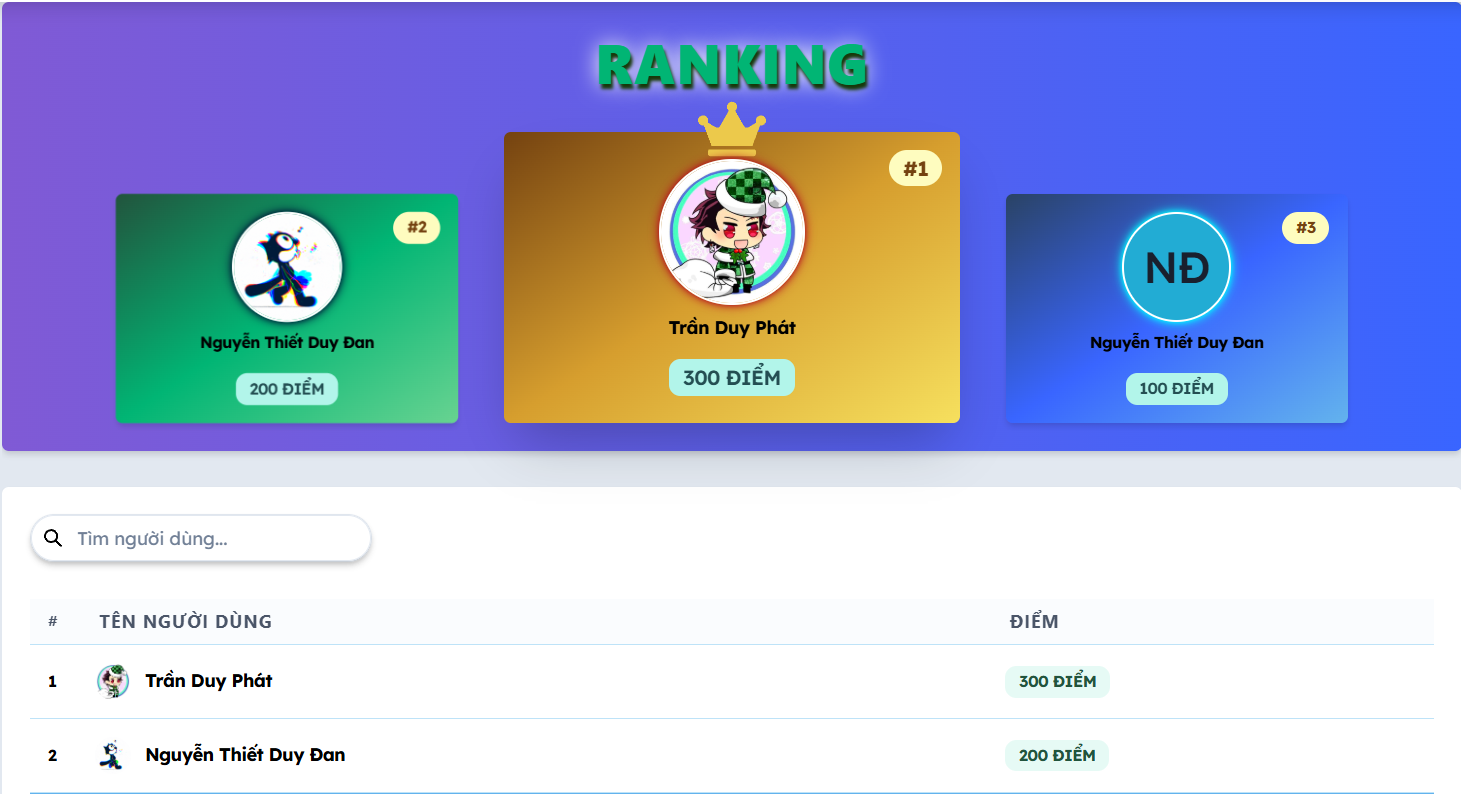
Điểm trong trang cá nhân

Thêm vào đó, khóa học cũng được cập nhật thêm tiến trình hoàn thành của một khóa học dựa trên những gì bạn vừa làm.



Hiển thị tiến trình hoàn thành

* + - 1. Giao diện xếp hạng

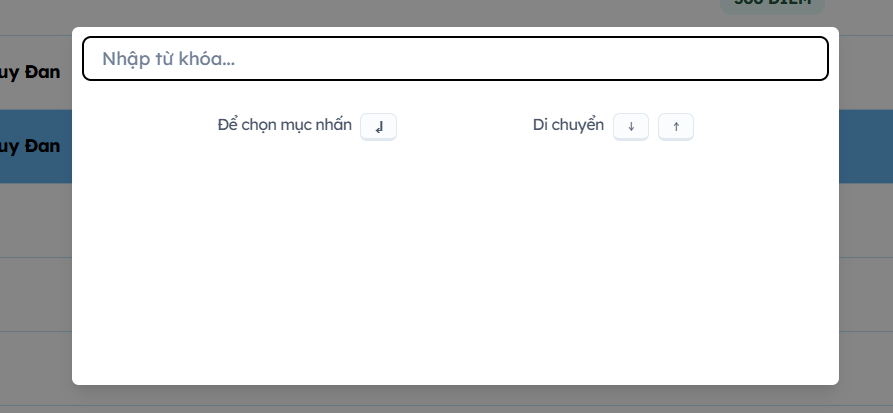


Giao diện xếp hạng

Ở giao diện này sẽ hiển thị top 3 người có điểm số cao nhất ở phần trên cùng, bên dưới sẽ là danh sách 100 người dùng có số điểm cao. Ta cũng có thể để top 5, top 10, top 50 đều được ở phần số dòng hiển thị. Ngoài ra còn có thêm một ô input để tìm kiếm người dùng và có thể bấm vào tên người dùng đó để xem được trang cá nhân người đó. Người dùng hiện tại sẽ được highlight nguyên dòng đó lên để người dùng có thể nhận diện được tài khoản của mình.

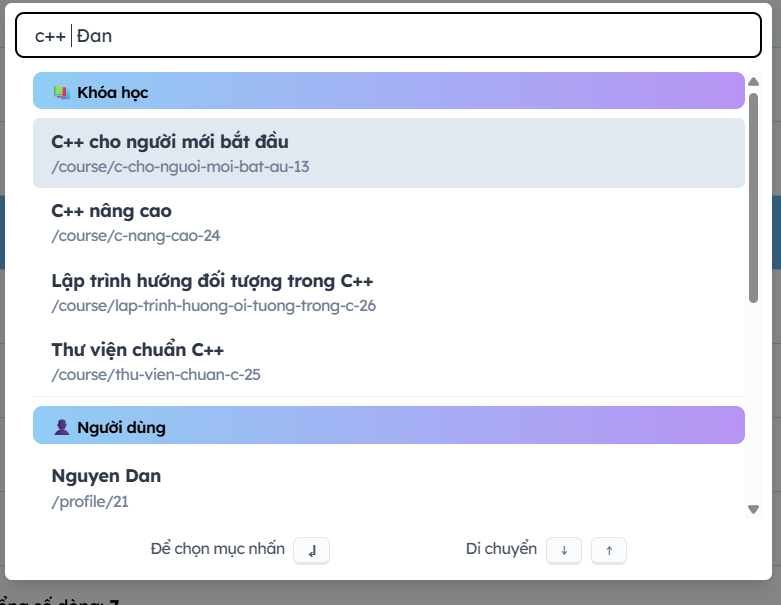
* + - 1. Giao diện tìm kiếm ở trang người dùng

Tìm kiếm ở trang người dùng có thể bấm vào ô tìm kiếm ở navbar hoặc cũng có thể dùng tổ hợp phím ctrl + / để mở lên modal tìm kiếm như sau.



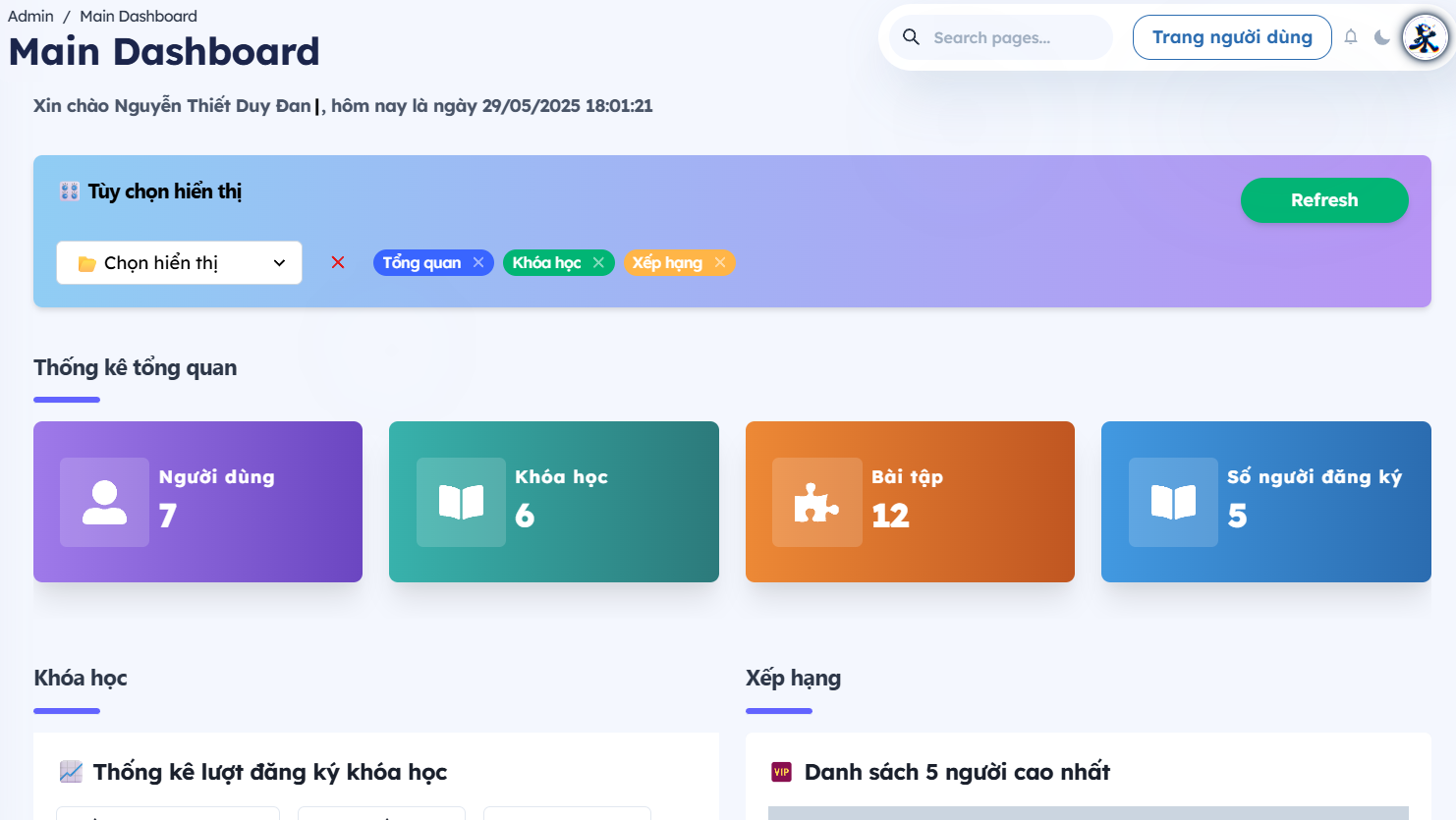
Modal tìm kiếm

Khi nhập từ khóa gợi ý sẽ xuất hiện cho người dùng lựa chọn và có 3 phần tìm kiếm ở đây đó là Khóa học, Người dùng và Bài viết. Mọi thứ đều được phần chia rõ ràng từng phần không xen lẫn vào nhau giúp cho người dùng có thể lựa chọn một cách dễ dàng. Tìm kiếm không cần phải nhấn enter hay bất kỳ nút nào để có thể ra kết quả.



Phần gợi ý khi nhập từ khóa

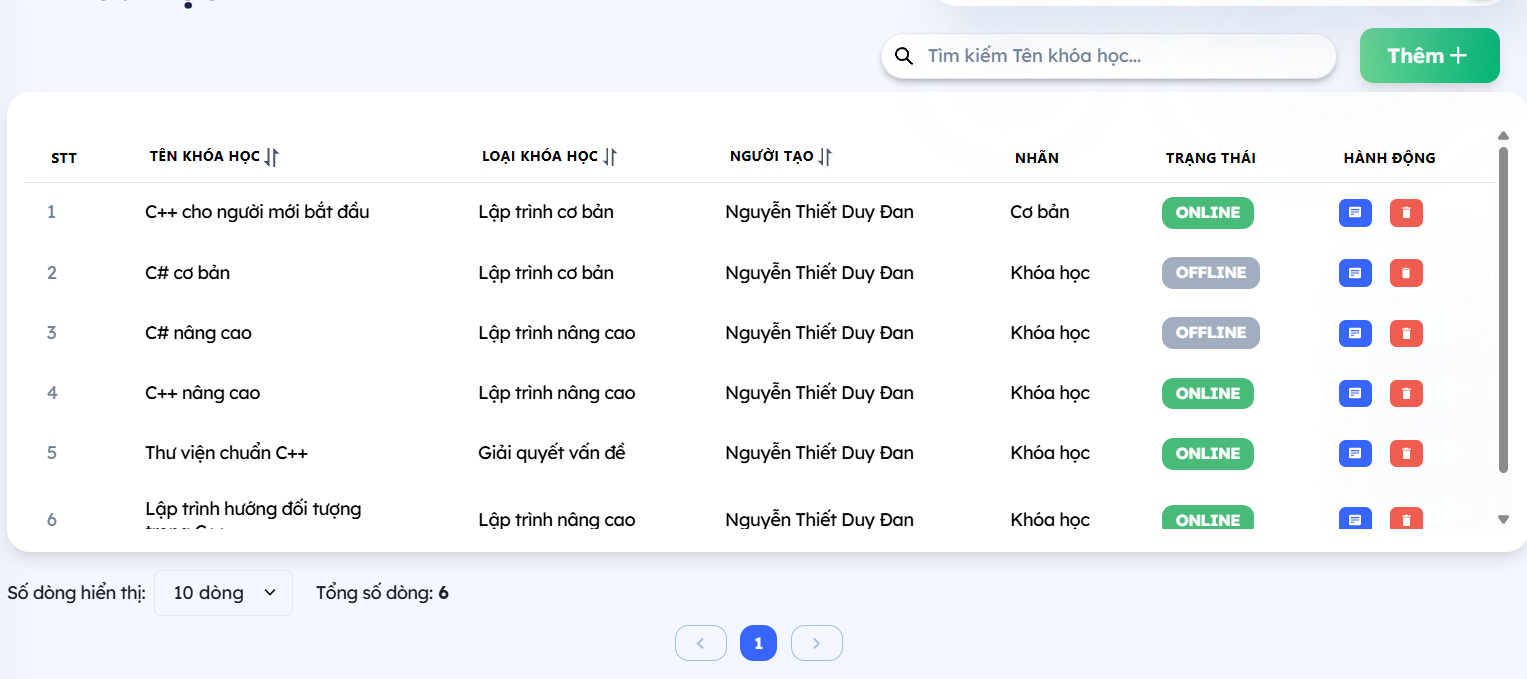
* + - 1. Giao diện Dashboard



Giao diện Dashboard

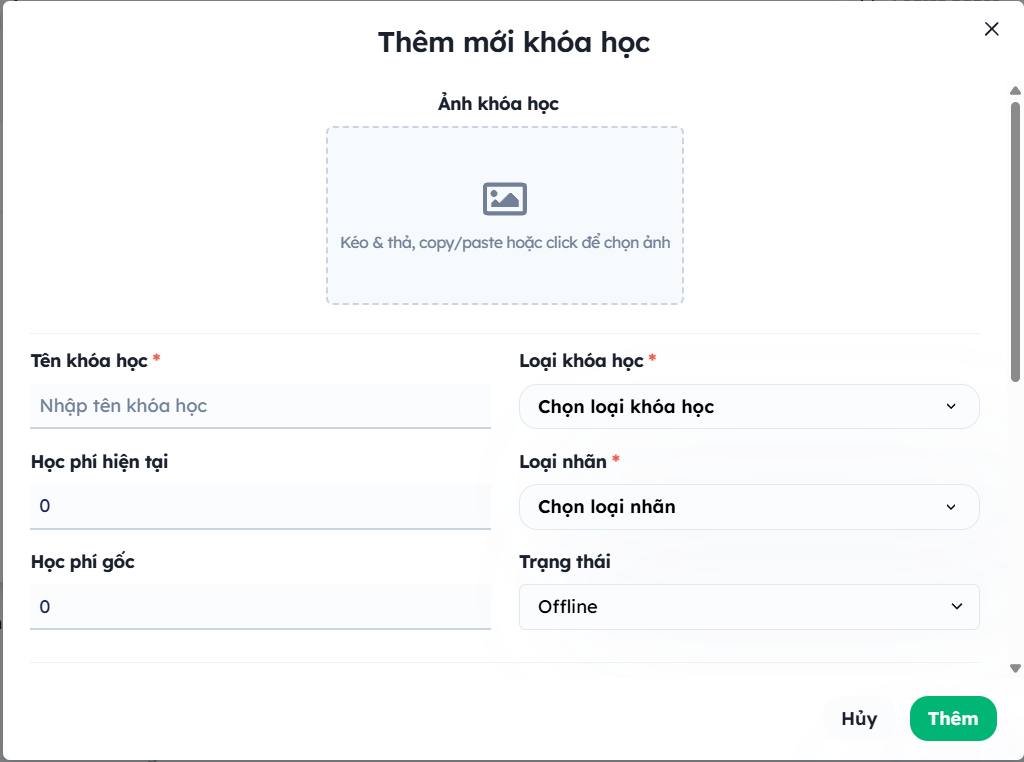
Giao diện này sẽ xuất hiện đầu tiên khi người dùng có quyền vào trang admin. Nó bao gồm các phần như sau:

* Phần các chức năng: Lựa chọn các phần có thể hiển thị như là tổng quan, khóa học, xếp hạng, … và có nút refresh để load lại dữ liệu khi cần.
* Bên dưới là phần hiển thị thống kê của các phần đã được lựa chọn ở trên.
  + - 1. Giao diện quản lý khóa học



Giao diện index của khóa học

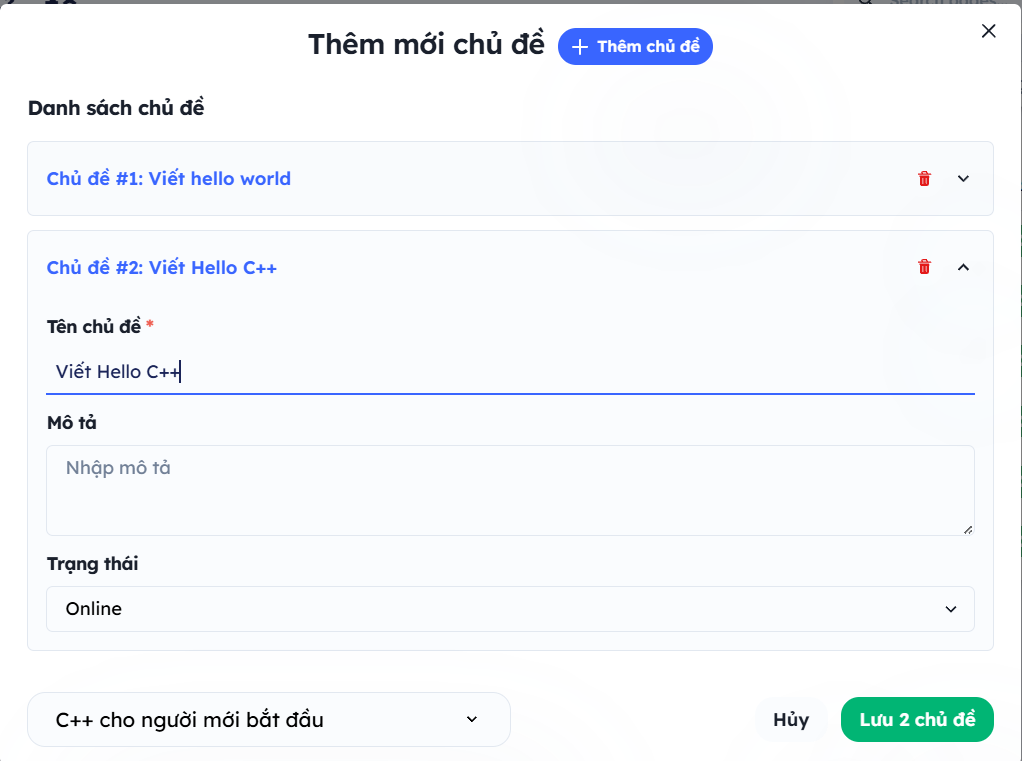
Ở giao diện nãy sẽ hiển thị danh sách khóa học, nút tìm kiếm, nút thêm khóa học mới và phần phân trang. Danh sách khóa học được được giới hạn trong một ô ở giữa, được phép scroll với phần header đưa đính lên đầu và không di chuyển.



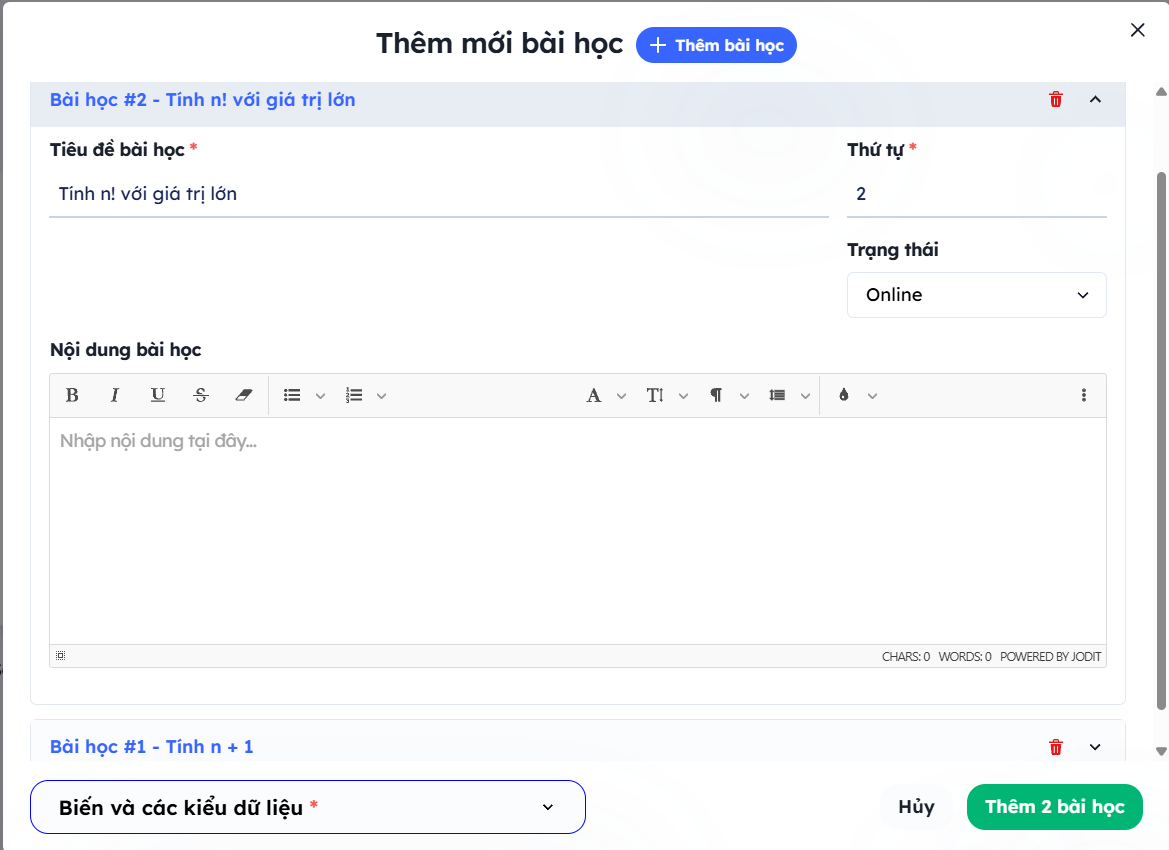
Giao diện nhập khóa học mới

* + - 1. Giao diện nhập một lúc nhiều dữ liệu của một số bảng

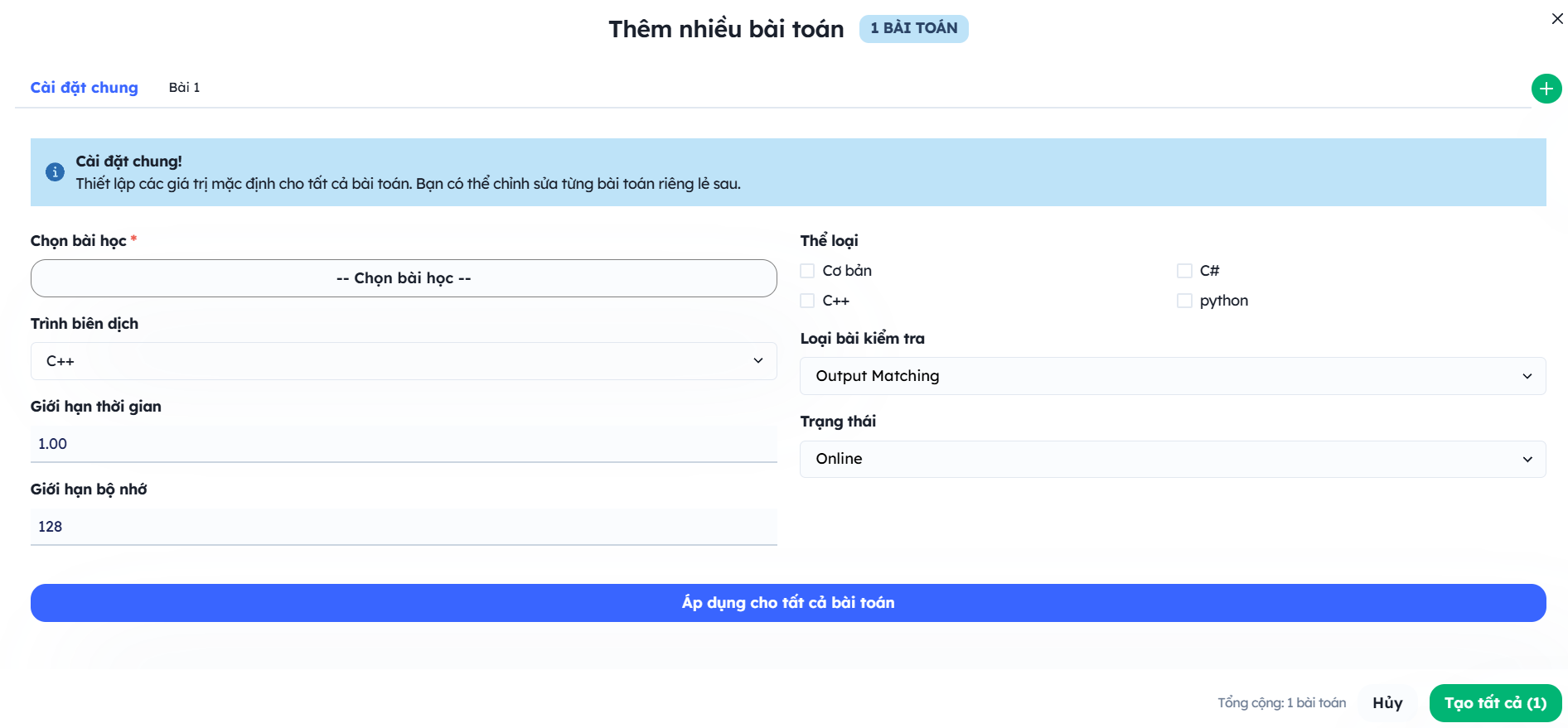
Với giao diện nhập một lúc nhiều dữ liệu này sẽ giúp cho người dùng có thể nhập một lúc nhiều dữ liệu mà không cần phải bật tắt nhiều modal.



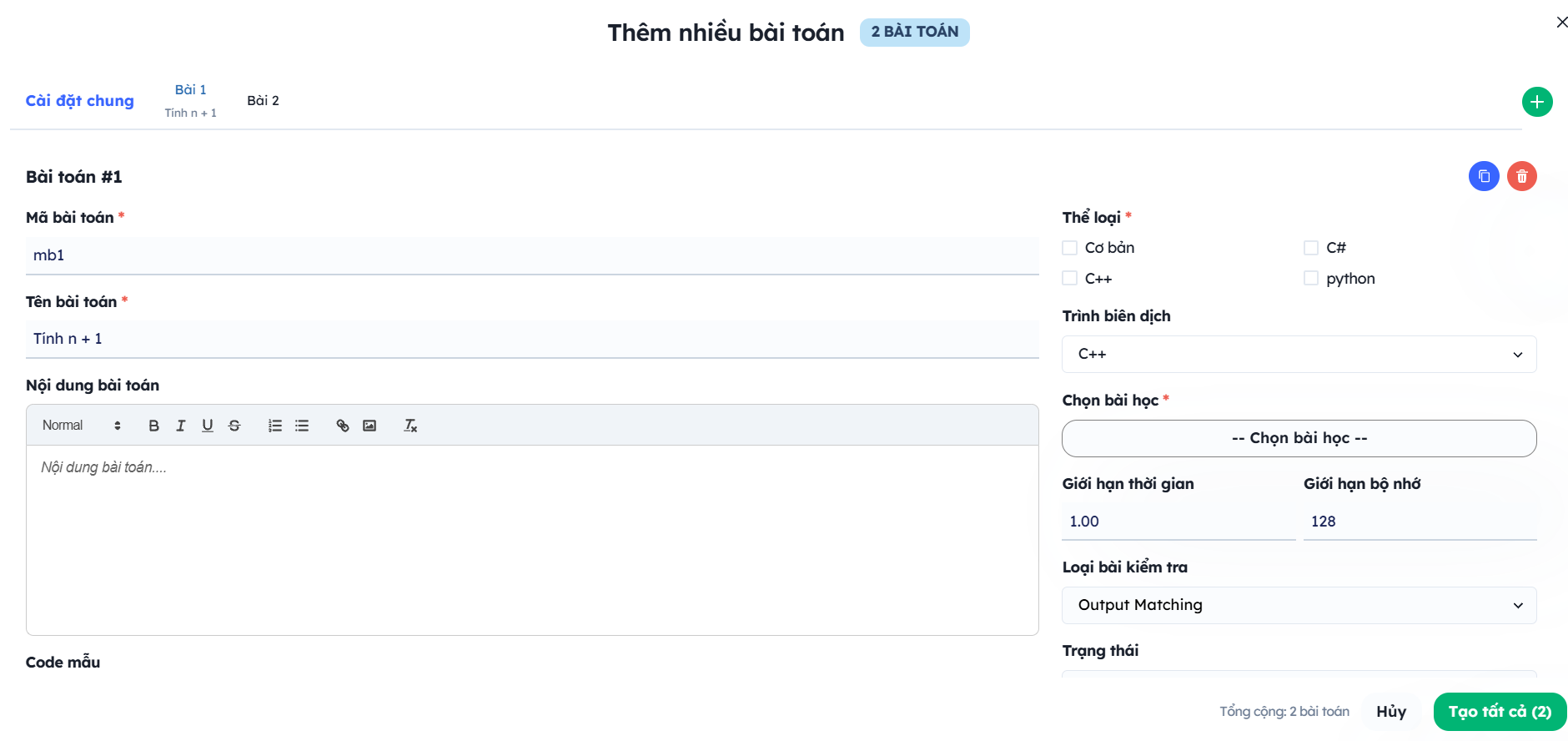
Giao diện nhập của chủ đề



Giao diện nhập của bài học



Giao diện nhập của bài toán (phần 1)



Giao diện nhập của bài toán (phần 2)

* 1. THẢO LUẬN
     1. Đánh giá hiệu quả và ưu điểm của giải pháp
        1. Đánh giá hiệu quả giải pháp

Sau quá trình nâng cấp và xây dựng lại hệ thống NTUCoder thành nền tảng **NTU-CODER-LMS**, đề tài đã đạt được nhiều thành quả mong đợi, có hiệu quả tích cực, góp phần khắc phục các hạn chế trên hệ thống cũ và nâng cao trải nghiệm người dùng. Việc đánh giá hiệu quả được dựa trên những khía cạnh sau: **giao diện người dùng, tính năng hệ thống, hiệu quả học tập, và khả năng mở rộng trong tương lai.** Qua việc khảo sát từ trải nghiệm từ những người dùng, đã có những phản hồi tích cực đến từ giao diện người dùng cũng như những chức năng khác như là code tại website hay là học tập trực tuyến. Giao diện được đánh giá khá tốt khi mà nó đã hiện đại hơn so với website cũ, trực quan, đẹp mắt và dễ dàng thao tác đối với người mới. Giao diện dành cho admin và giảng viên cũng được cải tiến rõ rệt khi mà các chức năng thuộc về cụm CRUD trở nên tiện lợi và nhanh chóng hơn. Hệ thống cũng tối ưu lại không có quá nhiều trang chuyển qua lại, chẳng hạn như là thêm hay xóa đều được thực hiện ngay tại trang chính. Hệ thống cũng có thêm tính xếp hạng để giúp cho người dùng tăng động lực học tập, nhằm góp phần nâng cao hiệu quả sử dụng hệ thống và làm đa dạng người dùng. Hiện tại, hệ thống chỉ mới chỉ hỗ trợ một ngôn ngữ là C/C++ nhưng vẫn có thể phát triển thêm. Vậy là về tổng quan, hệ thống hiện tại đã đáp ứng được những yêu cầu đặt ra lúc ban đầu về mặt chức năng lẫn giao diện người dùng.

Ngoài những phần về chức năng lẫn giao diện, hệ thống đã kết hợp giữa việc học tập trực tuyến và hệ thống chấm bài tự động giúp cho người dùng có lộ trình học rõ ràng, có thể theo dõi tiến độ cá nhân, đồng thời thúc đẩy tinh thần học tập thông qua cơ chế tính điểm và xếp hạng. Về khả năng mở rộng, hệ thống được xây dựng theo kiến trúc API rõ ràng, hỗ trợ tích hợp thêm các ngôn ngữ lập trình mới, tính năng học liệu đa dạng và khả năng phát triển thành nền tảng học lập trình quy mô lớn trong tương lai.

Vậy về tổng quan, hệ thống đã đáp ứng gần đầy đủ các yêu cầu về chức năng và giao diện mà ban đầu đề ra. Ngoài ra, về phần chia vai trò để quản lý cũng đã khá rõ ràng khi có 3 vai trò (quản trị - giảng viên – người dùng), đảm bảo sẽ thuận tiện và bảo mật trong quá trình quản lý hệ thống.

* + - 1. Ưu điểm của giải pháp

Sau quá trình thiết kế và phát triển hệ thống **NTU-CODER-LMS**, một số ưu điểm nổi bật của giải pháp đã được ghi nhận rõ rệt trong quá trình thử nghiệm:

* **Giao diện hiện đại, thân thiện với người dùng**: giao diện người dùng được thiết kế dựa trên sự kết hợp giữa **ReactJS + Vite và ChakraUI**, điều này đã mang lại cho người sử dụng trải nghiệm mượt mà cũng như là giúp cho người dùng dễ tiếp cận. Không chỉ sinh viên, giảng viên và quản trị viên điều có có giao diện riêng phù hợp với chức năng, từ đó cải thiện trải nghiệm hệ thống.
* **Hỗ trợ quản lý khóa học và bài học chuyên sâu:** Hệ thống hiện tại không chỉ dừng ở việc chấm bài tự động như NTUCoder trước đây, mà còn cho phép xây dựng các khóa học đầy đủ, bao gồm các chủ đề, bài học, bài tập tương tác và tiến độ học tập của từng người. Điều này giúp nâng cao chất lượng đào tạo, hỗ trợ cả người mới bắt đầu và người học nâng cao.
* **Tối ưu thao tác CRUD và luồng xử lý trên giao diện**: Các thao tác tạo, sửa, xóa dữ liệu như bài học, chủ đề, bài tập,... được thực hiện ngay tại giao diện chính, hạn chế việc chuyển trang không cần thiết. Ngoài ra, hệ thống còn hỗ trợ thêm nhiều dữ liệu cùng lúc, giúp tiết kiệm thời gian thao tác đáng kể cho giảng viên và quản trị viên.
* **Kiến trúc hệ thống rõ ràng, hỗ trợ mở rộng dễ dàng**: Hệ thống được phát triển theo mô hình API RESTful rõ ràng, dễ bảo trì và mở rộng trong tương lai. Việc bổ sung các ngôn ngữ lập trình khác (Java, Python, ...), hoặc tích hợp thêm các tính năng như diễn đàn trao đổi, quiz trắc nghiệm, ... hoàn toàn khả thi nhờ cấu trúc linh hoạt hiện tại.
* **Tăng cường khả năng tự học và theo dõi tiến độ học tập:** Mỗi người học có một trang cá nhân để theo dõi điểm số, phần trăm hoàn thành khóa học và lịch sử làm bài. Điều này giúp họ có cái nhìn toàn diện về tiến trình học tập và dễ dàng lên kế hoạch cải thiện năng lực cá nhân.
  + 1. Hạn chế của hệ thống hiện tại

Mặc dù hệ thống hiện tại đã đáp ứng gần đủ những yêu cầu đề ra nhưng bên cạnh đó vẫn còn tồn đọng một số lỗi cũng như hạn chế chưa được tối ưu. Thứ nhất, hiệu năng của việc tải trang vẫn chưa được tốt bởi vì là website sử dụng React đầu tay nên việc xử lý dữ liệu tải lần đầu chưa được tối ưu. Thứ hai, chức năng reponsive vẫn chưa được triển khai đầy đủ ở một số trang, vậy nên vẫn chưa được tối ưu ở một số thiết bị. Bên cạnh đó còn tồn đọng một số lỗi liên quan đến API được lấy ở ngoài chẳng hạn như của editor hoặc là của google Oauth2. Đôi lúc cũng sẽ xuất hiện một số lỗi liên quan đến nộp bài hoặc là lỗi liên quan đến kết nối.

Ngoài những lỗi ở trên thì vẫn còn hạn chế một chức năng, chẳng hạn như việc thêm khóa học từ file vẫn chưa được triển khai hoặc là export file cũng chưa được sử dụng. Hiện tại hệ thống vẫn chỉ chạy localhost vẫn chưa triển khai deloy rộng rãi, từ đó cũng không thể triển khai kiểm tra về lưu lượng truy cập, đó cũng là một hạn chế. Thêm vào đó còn một số chức năng vẫn còn tồn đọng ở BE mà vẫn chưa được triển khai. Hệ thống còn thiếu chức năng phân tích dữ liệu người dùng để đưa ra những đề xuất khóa học tốt nhất cho người dùng.

* + 1. Đề xuất hướng phát triển trong tương lai

Sau khi hoàn thiện giai đoạn xây dựng và nâng cấp hệ thống NTU-CODER-LMS, mặc dù sản phẩm đã đạt được những kết quả nhất định và đáp ứng phần lớn các yêu cầu ban đầu, nhưng vẫn còn nhiều hướng phát triển tiềm năng trong tương lai để nâng cao hiệu quả sử dụng cũng như mở rộng phạm vi ứng dụng trong thực tế. Hệ thống hiện tại vẫn chỉ ở mức thử nghiệm nội bộ, chưa được triển khai trên môi trường thực tế có nhiều người dùng đồng thời, do đó những đề xuất phát triển tiếp theo cần tập trung vào cả mặt kỹ thuật lẫn trải nghiệm người dùng để đưa hệ thống tiến xa hơn.

Trước hết, cần phải tối ưu hóa hiệu năng hệ thống, đặc biệt là ở các phần liên quan đến việc tải dữ liệu lần đầu. Do hệ thống được xây dựng bằng React và đây là một trong những sản phẩm đầu tay nên một số thao tác chưa được xử lý tối ưu. Hiện tượng tải trang chậm hoặc đơ nhẹ vẫn còn xảy ra, đặc biệt khi có nhiều thành phần phải gọi API cùng lúc. Trong tương lai, nhóm phát triển có thể áp dụng các kỹ thuật như lazy loading, memoization, sử dụng caching tại client, kết hợp cùng prefetching hợp lý để giảm thiểu độ trễ và tăng tốc độ phản hồi của giao diện. Đồng thời, cần kiểm soát kỹ hơn luồng dữ liệu trong React thông qua việc sử dụng các thư viện quản lý state hiệu quả như Redux Toolkit hoặc Zustand.

Tiếp theo, việc hoàn thiện giao diện responsive trên tất cả các thiết bị cũng là một yếu tố quan trọng. Trong thời đại học tập đa nền tảng, học viên có thể học trên điện thoại, máy tính bảng, hoặc laptop ở bất cứ đâu. Tuy nhiên, hiện tại vẫn còn một số trang chưa hiển thị tốt trên thiết bị di động. Do đó, trong các bản nâng cấp tiếp theo, cần thiết kế lại phần giao diện theo hướng mobile-first, đồng thời kiểm tra trải nghiệm người dùng thực tế trên nhiều kích thước màn hình khác nhau để đảm bảo tính linh hoạt và thân thiện.

Ngoài ra, hệ thống cũng nên mở rộng các chức năng hỗ trợ quản lý như import và export dữ liệu từ các file định dạng CSV hoặc Excel. Chức năng này sẽ giúp giảng viên hoặc quản trị viên tiết kiệm thời gian khi thêm số lượng lớn bài học, chủ đề hoặc danh sách học viên. Việc xuất file cũng sẽ hỗ trợ quá trình lưu trữ dữ liệu hoặc trích xuất thống kê học tập theo lớp, theo người dùng. Đây là những chức năng rất cần thiết đối với các hệ thống quản lý đào tạo chuyên nghiệp.

Thêm vào đó, một tính năng quan trọng khác cần được triển khai trong tương lai là khả năng phân tích hành vi học tập của người dùng. Việc thu thập dữ liệu người dùng và xử lý phân tích bằng các thuật toán đơn giản sẽ giúp hệ thống có thể đề xuất các khóa học phù hợp với nhu cầu và trình độ hiện tại của từng người dùng, từ đó tăng khả năng cá nhân hóa nội dung học và giữ chân người học tốt hơn. Các mô-đun phân tích học tập (learning analytics) có thể được tích hợp để đánh giá hiệu suất, thời gian học, số lần hoàn thành bài, v.v.

Cuối cùng, một bước phát triển quan trọng chính là việc triển khai hệ thống trên môi trường thực tế thay vì chỉ chạy cục bộ (localhost). Việc đưa hệ thống lên các nền tảng hosting hoặc cloud (như Vercel, Firebase, hoặc VPS riêng) không những giúp kiểm tra độ ổn định khi có nhiều người truy cập cùng lúc mà còn tạo điều kiện để mời người dùng trải nghiệm thực tế và đóng góp ý kiến cải thiện. Đồng thời, triển khai thật sẽ giúp phát hiện các lỗi phát sinh trong môi trường thực tế mà trong quá trình phát triển khó nhận ra.

Tóm lại, hệ thống NTU-CODER-LMS hoàn toàn có tiềm năng mở rộng mạnh mẽ trong thời gian tới, không chỉ ở mặt kỹ thuật mà còn ở mặt chiến lược phát triển sản phẩm. Với sự đầu tư đúng đắn và tiếp thu phản hồi người dùng, hệ thống sẽ ngày càng hoàn thiện, góp phần phục vụ tốt hơn cho quá trình học tập và giảng dạy lập trình trong môi trường đại học cũng như cộng đồng lập trình nói chung.

1. KẾT LUẬN VÀ KIẾN LUẬN
   1. KẾT LUẬN

Sau một quá trình tìm hiểu, phát triển và thử nghiệm, em nhận thấy việc nâng cấp và xây dựng hệ thống NTU-CODER-LMS từ nền tảng NTUCoder cũ là một bước đi đúng đắn và cần thiết trong bối cảnh hiện nay khi nhu cầu học lập trình và đánh giá kết quả học tập thông qua các hệ thống trực tuyến ngày càng tăng. Trong suốt quá trình thực hiện đề tài, em đã cố gắng nghiên cứu và ứng dụng những công nghệ mới cũng như tối ưu lại luồng xử lý của hệ thống để giúp người dùng có trải nghiệm tốt hơn. Kết quả cho thấy hệ thống mới không những đáp ứng được những chức năng cơ bản như tạo khóa học, tạo bài tập, chấm điểm tự động mà còn được mở rộng với giao diện trực quan hơn, hiện đại hơn và dễ sử dụng hơn so với phiên bản cũ. Đặc biệt, việc xây dựng chức năng xếp hạng người dùng sau mỗi tuần cũng đã tạo ra sự cạnh tranh lành mạnh và khuyến khích người học cố gắng hơn trong việc hoàn thành các bài tập. Điều này phần nào cho thấy hiệu quả của việc cải tiến không chỉ nằm ở mặt kỹ thuật mà còn ảnh hưởng tích cực đến tâm lý và động lực học tập. Dù vẫn còn những điểm chưa hoàn thiện như chưa tích hợp đầy đủ responsive cho giao diện hay chưa hỗ trợ nhiều ngôn ngữ lập trình, nhưng với những gì đã đạt được, em tin rằng hệ thống đã hoàn thành tốt các mục tiêu đề ra trong đề tài, đồng thời mở ra nhiều hướng đi mới để phát triển sâu hơn trong tương lai.

* 1. KIẾN LUẬN

Trong quá trình xây dựng và thử nghiệm hệ thống NTU-CODER-LMS, em nhận ra rằng một hệ thống học lập trình và chấm bài tự động như thế này không thể dừng lại ở một phiên bản cố định mà cần được liên tục cập nhật, bổ sung và tối ưu hóa để theo kịp nhu cầu thực tế của người dùng. Do đó, em xin đề xuất một số hướng phát triển tiếp theo để nâng cao chất lượng cũng như mở rộng phạm vi sử dụng của hệ thống. Trước tiên là về mặt kỹ thuật, hệ thống hiện tại mới chỉ chạy ở môi trường localhost, vì vậy trong tương lai cần triển khai hệ thống lên server thật hoặc nền tảng cloud để có thể đánh giá hiệu năng trong điều kiện thực tế, đồng thời dễ dàng mở rộng số lượng người dùng. Bên cạnh đó, việc bổ sung thêm các tính năng như import, export dữ liệu khóa học bằng file, phân tích hành vi người học hay đưa ra gợi ý bài học phù hợp sẽ giúp hệ thống trở nên thông minh và hữu ích hơn, nhất là đối với người học mới bắt đầu. Hơn nữa, để phục vụ nhu cầu đa dạng của người học, hệ thống nên được mở rộng thêm các ngôn ngữ lập trình phổ biến khác như Python, Java hoặc JavaScript thay vì chỉ hỗ trợ C/C++. Việc triển khai giao diện tương thích với nhiều thiết bị, từ điện thoại đến máy tính bảng cũng sẽ là một điểm cộng lớn giúp tăng khả năng tiếp cận cho hệ thống. Về mặt triển khai thực tế, em rất mong nhà trường hoặc bộ phận chuyên trách CNTT sẽ cân nhắc sử dụng hệ thống này như một công cụ hỗ trợ giảng dạy trong các học phần lập trình. Đây không chỉ là cơ hội để sinh viên có trải nghiệm học tập hiện đại hơn mà còn là bước đệm để cải tiến chất lượng đào tạo theo hướng ứng dụng công nghệ thông tin hiệu quả. Nếu có thể được đầu tư thêm về nguồn lực và thời gian, em tin rằng hệ thống NTU-CODER-LMS hoàn toàn có thể trở thành một nền tảng học lập trình có tính ứng dụng thực tế cao, không chỉ trong phạm vi trường đại học mà còn mở rộng đến cộng đồng học lập trình rộng lớn hơn.

TÀI LIỆU THAM KHẢO

1. Nguyễn Đức Thuần (2007), *Giáo trình Cơ sở dữ liệu*, Khoa Công Nghệ Thông Tin (NTU)
2. **Wikipedia,** *REST,* <https://en.wikipedia.org/wiki/REST>
3. Lương Thị Thảo Hiếu, Lê Thanh Của và Đào Phương Anh (2018). *Tích hợp restful web service trên nền tảng di động xây dựng hệ thống web khoa công nghệ thông tin trên điện thoại android*. Tạp chí KHCN, số 09/2018, tr. 45-50.
4. Nguyễn Thị Yến Hợp (2017), *Nghiên cứu về web api và viết ứng dụng xây dựng phần mềm quản lý bảo hành thiết bị tin học trên web service,* Khoá luận tốt nghiệp đại học.
5. **viblo.asia,** *RESTful API: Kiến thức thiết yếu cho Backend Developers (phần 1),* https://viblo.asia/p/restful-api-kien-thuc-thiet-yeu-cho-backend-developers-phan-1-MG24Bdd5Lz3
6. **aptech.fpt.edu.vn**, *ReactJS là gì? Những điều bạn chưa biết về thư viện mã nguồn mở này.,* <https://aptech.fpt.edu.vn/reactjs-la-gi.html>
7. **docs.docker.com,** *What is Docker?,* https://docs.docker.com/get-started/docker-overview
8. **viblo.asia**, *Minio – Object storage server như AWS S3,* https://viblo.asia/p/minio-object-storage-server-nhu-aws-s3-LzD5d0AW5jY
9. Chakra Systems, CharkaUI, https://chakra-ui.com/