

ĐẠI HỌC BÁCH KHOA HÀ NỘI

NGHIÊN CỨU TỐT NGHIỆP 2

**Phát triển website soạn thảo và biên dịch tài liệu
LaTeX**

NGUYỄN HỮU ĐỨC

DUC.NH210192@sis.hust.edu.vn

Ngành Công nghệ thông tin và truyền thông

Giảng viên hướng dẫn: TS. Nguyễn Đình Thuận

Khoa: Kỹ thuật máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 01/2025

TÓM TẮT NỘI DUNG

Việc soạn thảo và biên dịch tài liệu LaTeX hiện nay vẫn còn nhiều hạn chế về mặt cài đặt, khả năng cộng tác và tùy biến. Mặc dù các công cụ trực tuyến như Overleaf đã giải quyết một phần vấn đề, nhưng chúng vẫn còn bị giới hạn bởi các tính năng, tùy chọn cấu hình và chi phí. Hơn nữa, việc phụ thuộc vào các dịch vụ bên thứ ba cũng tiềm ẩn những rủi ro về tính khả dụng và bảo mật.

Để khắc phục những hạn chế trên, chúng tôi đã phát triển một trang web soạn thảo và biên dịch LaTeX. Giải pháp này cho phép người dùng tải lên các file LaTeX và các tập tin liên quan, sau đó hệ thống sẽ tiến hành biên dịch và trả về kết quả dưới dạng file PDF. Trang web được xây dựng trên nền tảng fullstack, kết hợp giữa VueJS ở frontend và ASP.NET Core Web API ở backend.

Đóng góp chính của đề án là việc xây dựng một công cụ soạn thảo và biên dịch LaTeX trực tuyến, đáp ứng được nhu cầu sử dụng của các nhóm nghiên cứu, sinh viên và giảng viên. Trang web không chỉ hỗ trợ quá trình soạn thảo và biên dịch mà còn tạo điều kiện cho việc cộng tác giữa nhiều người dùng. Kết quả cuối cùng là một sản phẩm phần mềm ổn định, có thể tùy biến và đáp ứng được các yêu cầu của người dùng.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp.....	1
CHƯƠNG 2. TỔNG QUAN CHỨC NĂNG.....	2
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	3
3.1 Frontend	3
3.1.1 JavaScript.....	3
3.1.2 Vue.js	3
3.1.3 Ant Design	3
3.1.4 Monaco Editor.....	3
3.2 Backend.....	3
3.2.1 C#	3
3.2.2 ASP.NET Core	3
3.2.3 Entity Framework Core	3
3.2.4 SQL Server	4
3.3 Các công nghệ khác	4
CHƯƠNG 4. TRIỂN KHAI HỆ THỐNG.....	5
4.1 Thư viện và công cụ sử dụng	5
4.2 Triển khai	5
4.2.1 Frontend.....	5
4.2.2 Backend	5
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	7
5.1 Kết luận.....	7

5.2 Hướng phát triển.....	7
---------------------------	---

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong thời đại số hóa, việc soạn thảo và chia sẻ tài liệu khoa học kỹ thuật ngày càng trở nên phổ biến. LaTeX, với cú pháp chính xác và khả năng tạo ra các tài liệu chất lượng cao, đã trở thành một công cụ không thể thiếu cho các nhà nghiên cứu. Tuy nhiên, quá trình soạn thảo và biên dịch LaTeX truyền thống thường phức tạp và mất thời gian. Các công cụ trực tuyến hiện có, dù đã cải thiện trải nghiệm người dùng, vẫn còn nhiều hạn chế về tính năng và khả năng tùy biến. Yêu cầu đặt ra là tạo một môi trường soạn thảo và biên dịch LaTeX thuận tiện cho người dùng.

1.2 Mục tiêu và phạm vi đề tài

Mục tiêu là phát triển một ứng dụng web LaTeX trực tuyến. Ứng dụng sẽ hỗ trợ đầy đủ các tính năng soạn thảo và biên dịch LaTeX, cho phép người dùng cộng tác trực tuyến, và tích hợp với các dịch vụ khác. Phạm vi nghiên cứu tập trung vào việc xây dựng một nền tảng ổn định, dễ sử dụng, phục vụ cho nhu cầu của sinh viên, nghiên cứu sinh và các nhà khoa học.

1.3 Định hướng giải pháp

Để giải quyết vấn đề trên, chúng tôi đề xuất phát triển một ứng dụng web LaTeX dựa trên kiến trúc fullstack, kết hợp giữa VueJS cho frontend và ASP.NET Core API cho backend. Việc lựa chọn VueJS là do khả năng xây dựng giao diện người dùng động, trong khi ASP.NET Core cung cấp một nền tảng phát triển các ứng dụng web backend. Đóng góp chính của đề án là việc tạo ra một môi trường làm việc trực quan, thân thiện với người dùng, đồng thời cung cấp các tính năng cộng tác thời gian thực.

CHƯƠNG 2. TỔNG QUAN CHỨC NĂNG

Use Case	Mô tả
Đăng ký	Tạo một tài khoản mới để sử dụng hệ thống.
Đăng nhập	Xác thực thông tin đăng nhập để truy cập hệ thống.
Đăng xuất	Kết thúc phiên làm việc hiện tại.
Xem danh sách dự án	Hiển thị danh sách các dự án mà người dùng đang tham gia.
Tạo dự án mới	Tạo một dự án mới với thông tin cơ bản.
Sao chép dự án	Tạo một bản sao của dự án có sẵn.
Xóa dự án	Xóa một dự án khỏi hệ thống.
Cập nhật chế độ hiển thị	Cập nhật chế độ hiển thị của dự án, cho phép người dùng ngoài truy cập dự án bằng đường dẫn.
Thêm thành viên cộng tác	Mời người khác tham gia vào dự án và cài đặt quyền hạn đối với dự án.
Soạn thảo LaTeX	Viết, chỉnh sửa tài liệu LaTeX.
Biên dịch LaTeX	Biên dịch dự án LaTeX thành định dạng PDF.
Lưu file	Lưu lại những thay đổi của file lên hệ thống.
Thêm file	Thêm file tài liệu vào dự án.
Xóa file	Xóa file tài liệu khỏi dự án.
Lưu phiên bản dự án	Lưu lại trạng thái hiện tại của dự án thành một phiên bản mới.
Xem phiên bản dự án	Xem lịch sử các phiên bản của dự án.
Phục hồi phiên bản dự án	Khôi phục dự án về một phiên bản cũ.

Danh sách các use case của người dùng

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

3.1 Frontend

Frontend của ứng dụng được xây dựng dựa trên các công nghệ sau:

3.1.1 JavaScript

JavaScript là ngôn ngữ lập trình cốt lõi để tạo ra các tương tác động trên trang web. Trong dự án này, JavaScript được sử dụng để:

- Xây dựng các hiệu ứng động.
- Xử lý các sự kiện của người dùng.
- Cập nhật giao diện người dùng.

3.1.2 Vue.js

Vue.js là một framework JavaScript giúp đơn giản hóa việc xây dựng các giao diện người dùng. VueJS được sử dụng để xây dựng cấu trúc giao diện chính, quản lý thành phần và dữ liệu.

3.1.3 Ant Design

Ant Design là một thư viện UI cung cấp một bộ các thành phần giao diện người dùng; sử dụng để xây dựng các thành phần giao diện như: form, button, table, modal.

3.1.4 Monaco Editor

Monaco Editor là một editor code được phát triển bởi Microsoft; cung cấp một môi trường soạn thảo code LaTeX.

3.2 Backend

Backend của ứng dụng được xây dựng dựa trên các công nghệ sau:

3.2.1 C#

C# là ngôn ngữ lập trình hướng đối tượng mạnh mẽ, được sử dụng để viết các logic nghiệp vụ của ứng dụng.

3.2.2 ASP.NET Core

ASP.NET Core là một framework web mã nguồn mở, đa nền tảng của Microsoft; sử dụng để xây dựng API phục vụ các yêu cầu từ frontend.

3.2.3 Entity Framework Core

Entity Framework Core là một ORM giúp ánh xạ dữ liệu giữa các đối tượng trong ứng dụng với cơ sở dữ liệu; sử dụng để quản lý dữ liệu người dùng, tài liệu

LaTeX, cấu hình ứng dụng.

3.2.4 SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ của Microsoft; sử dụng để lưu trữ các dữ liệu quan trọng của ứng dụng.

3.3 Các công nghệ khác

- JSON Web Token (JWT): Sử dụng để xác thực người dùng và trao đổi thông tin an toàn.
- WebSocket: Công nghệ giao tiếp hai chiều, dùng để cập nhật dữ liệu thời gian thực.

CHƯƠNG 4. TRIỂN KHAI HỆ THỐNG

4.1 Thư viện và công cụ sử dụng

Mục đích	Công cụ	Địa chỉ URL
Lập trình frontend	NodeJS v18.20.6	https://nodejs.org/
Lập trình backend	ASP.NET Core 8	https://dotnet.microsoft.com/
Cơ sở dữ liệu	SQL Server 2019	https://www.microsoft.com/en-us/sql-server/
Biên dịch LaTeX	TeX Live	https://tug.org/texlive/

Danh sách thư viện và công cụ sử dụng

4.2 Triển khai

4.2.1 Frontend

1. Cài đặt NodeJS theo hướng dẫn.
2. Mở mã nguồn frontend, cập nhật biến môi trường trong file `.env` tương tự `.env.example`
3. Chạy lệnh cài đặt các gói cần thiết: `'npm install'`
4. Chạy ứng dụng ở môi trường develop: `'npm run dev'`
5. Build ứng dụng thông qua lệnh `'npm run build'`. Khởi động một máy chủ cục bộ để chạy ứng dụng trên môi trường phát triển: `'npm run previews'`

4.2.2 Backend

a, Trình biên dịch TeXLive

Trong phần này, chúng ta sẽ cài đặt các công cụ cần thiết để biên dịch tài liệu LaTeX. TeX Live là một phân phối LaTeX bao gồm hầu hết các gói lệnh cần thiết để tạo ra các tài liệu chất lượng cao. Việc cài đặt đầy đủ các gói lệnh sẽ giúp bạn có thể tạo ra nhiều loại tài liệu khác nhau, từ các bài báo khoa học đến các cuốn sách.

Cài đặt trình biên dịch TeXLive theo hướng dẫn.

Nếu chạy trên hệ điều hành dựa trên Debian như Ubuntu, có thể cài đặt qua apt:

- Cài đặt toàn bộ: `sudo apt install texlive-full`
- Cài đặt cơ bản: `sudo apt install texlive-base`

b, Cơ sở dữ liệu

1. Cài đặt SQL Server theo hướng dẫn.
2. Nếu gặp khó khăn trong quá trình cài đặt trực tiếp (ví dụ, SQL Server không hỗ trợ Ubuntu 22.04), có thể cân nhắc sử dụng Docker để triển khai SQL

Server trong một container.

- Tham khảo tài liệu chính thức để cài đặt Docker.
- Tạo file docker-compose.yml. Nếu đã có file này, hãy kiểm tra lại nội dung file.
- Mở terminal, điều hướng đến thư mục chứa file 'docker-compose.yml' và thực hiện lệnh: 'docker-compose up -d'
Lệnh này sẽ tạo và khởi động một container SQL Server theo cấu hình trong file YAML.

c, ASP.NET Core

1. Cài đặt .NET SDK (hệ thống đang sử dụng .NET 8) theo hướng dẫn.
Nếu chạy trên hệ điều hành dựa trên Debian như Ubuntu, có thể cài đặt qua apt: 'sudo apt install dotnet-sdk-8.0'
2. Khôi phục các gói cần thiết: 'dotnet restore'
3. Mở appsettings.json và cập nhật nếu cần thiết (VD: cập nhật Connection-Strings là thông tin kết nối cơ sở dữ liệu)
4. Cài đặt công cụ EF Core: 'dotnet tool install --global dotnet-ef'
5. Tạo hoặc cập nhật cơ sở dữ liệu: 'dotnet ef database update'
Lệnh này sẽ tự động tạo hoặc cập nhật cơ sở dữ liệu dựa trên các model đã định nghĩa trong dự án.
6. Biên dịch: 'dotnet build'
7. Chạy ứng dụng: 'dotnet run'
Ứng dụng sẽ bắt đầu chạy và lắng nghe trên cổng mặc định.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Trong quá trình thực hiện, tôi đã kế thừa và phát triển hệ thống soạn thảo và chia sẻ tài liệu LaTeX từ đồ án của chị Trần Thanh Hiền. Tôi đã tập trung vào việc mở rộng khả năng tương thích của hệ thống lên nền tảng Linux, đồng thời bổ sung cơ chế xử lý xung đột giữa các phiên bản khi nhiều người cùng cập nhật tài liệu. Tính năng này giúp đảm bảo tính nhất quán và tránh mất dữ liệu. Thông qua việc xử lý xung đột phiên bản, người dùng có thể dễ dàng so sánh và quyết định những thay đổi cần giữ lại, từ đó nâng cao hiệu quả làm việc nhóm.

5.2 Hướng phát triển

- Tích hợp với các công cụ khác: Tích hợp các công cụ sao lưu trên đám mây như OneDrive.
- Trí tuệ nhân tạo: Áp dụng để tự động hóa một số tác vụ, như đề xuất sửa lỗi chính tả, gợi ý nội dung.