BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**🙞 🕮 🙜**



**NIÊN LUẬN**

**NGÀNH KHOA HỌC MÁY TÍNH**

**Đề tài**

**XÂY DỰNG WEBSITE QUẢN LÝ CÔNG VĂN**

**TÍCH HỢP NHẬN DẠNG VĂN BẢN**

**Sinh viên thực hiện: Hồ Minh Quân**

**Mã số: B2017073**

**Khóa: K46**

***Cần Thơ, 05/2024***

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**🙞 🕮 🙜**



**NIÊN LUẬN NGÀNH**

**KHOA HỌC MÁY TÍNH**

**Đề tài**

**XÂY DỰNG WEBSITE QUẢN LÝ CÔNG VĂN**

**TÍCH HỢP NHẬN DẠNG VĂN BẢN**

**Giảng Viên hướng dẫn Sinh viên thực hiện**

**Th.S Phạm Xuân Hiền Hồ Minh Quân**

**Mã số: B2017073**

**Khóa: K46**

***Học kỳ 2, 2023 - 2024***

***Cần Thơ, 05/2024***

**NHẬN XÉT CỦA GIẢNG VIÊN**

**--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

**LỜI CẢM ƠN**

*Lời đầu tiên, em xin chân thành cảm ơn quí thầy cô đã giúp em thực hiện đề tài này. Đặc biệt, là cô* ***Phạm Xuân Hiền*** *đã tận tình hướng dẫn, giúp đỡ, chỉ bảo em trong suốt khoảng thời gian làm niên luận.*

*Đồng thời em cũng xin trân trọng cảm ơn những tình cảm quý báu mà các thầy cô trong trường Đại học Cần Thơ đã truyền đạt cho em, những kinh nghiệm, kỹ thuật và cách thức trong việc xây dựng đề tài.*

*Cuối cùng em xin được gửi lời cảm ơn đến gia đình đã luôn động viên, ủng hộ, những người bạn đã gắn bó, chia sẻ rất nhiều kinh nghiệm, kiến thức và nhất là trong khoảng thời gian thực hiện đề tài, để đề tài có thể hoàn thành một cách thành công nhất.*

*Tuy nhiên do thời gian có hạn nên em không thể phát huy hết những ý tưởng khả năng hỗ trợ của ngôn ngữ và kỹ thuật lập trình vào đề tài, trong quá trình xây dựng website không thể tránh khỏi những sai sót, mong nhận được sự góp ý và thông cảm của quý thầy cô và các bạn.*

*Em xin chân thành cảm ơn.*

***Cần Thơ****, ngày 10 tháng 05 năm 2024*

**Sinh viên thực hiện**

**Hồ Minh Quân**

**MỤC LỤC**

[PHẦN GIỚI THIỆU 7](#_Toc166195153)

[1. Đặt vấn đề. 7](#_Toc166195154)

[2. Lịch sử giải quyết vấn đề. 7](#_Toc166195155)

[3. Mục tiêu đề tài. 8](#_Toc166195156)

[4. Đối tượng và phạm vi nghiên cứu của đề tài. 8](#_Toc166195157)

[5. Nội dung nghiên cứu. 9](#_Toc166195158)

[6. Kết quả đạt được. 9](#_Toc166195159)

[7. Bố cục báo cáo. 9](#_Toc166195160)

[PHẦN NỘI DUNG 10](#_Toc166195161)

[CHƯƠNG 1: MÔ TẢ BÀI TOÁN 10](#_Toc166195162)

[1. Mô tả chi tiết bài toán 10](#_Toc166195163)

[2. Vấn đề và giải pháp liên quan đến bài toán 10](#_Toc166195164)

[2.1. Xây dựng và huấn luyện mô hình 10](#_Toc166195165)

[3. Mô tả giải pháp cho bài toán 11](#_Toc166195166)

[3.1. Django 11](#_Toc166195167)

[3.2. MongoDB 12](#_Toc166195168)

[3.3. Phương pháp nhận dạng văn bản 13](#_Toc166195169)

[CHƯƠNG 2: THIẾT KẾ VÀ CÀI ĐẶT 17](#_Toc166195170)

[1. Thiết kế hệ thống 17](#_Toc166195171)

[1.1. Các chức năng của website 17](#_Toc166195172)

[1.2. Cơ sở dữ liệu 17](#_Toc166195173)

[2. Thiết kế và cài đặt giải thuật 18](#_Toc166195174)

[2.1. Nhận diện số công văn 18](#_Toc166195175)

[2.2. Nhận diện ngày ban hành công văn 19](#_Toc166195176)

[2.3. Nhận dạng email 19](#_Toc166195177)

[2.4. Nhận dạng tên cơ quan ban hành 20](#_Toc166195178)

[2.5. Phân loại lĩnh vực công văn 20](#_Toc166195179)

[3. Đánh giá mô hình 21](#_Toc166195180)

[3.1. Đánh giá độ chính xác của từng mô hình 21](#_Toc166195181)

[3.2. So sánh độ hiệu quả của các mô hình 22](#_Toc166195182)

[CHƯƠNG 3: KIỂM THỬ VÀ ĐÁNH GIÁ 23](#_Toc166195183)

[1. Mục tiêu 23](#_Toc166195184)

[2. Giao diện và chức năng 23](#_Toc166195185)

[2.1. Giao diện trang chủ 24](#_Toc166195186)

[2.2. Giao diện trang quản lý công văn 24](#_Toc166195187)

[2.3. Giao diện trang tìm kiếm công văn 25](#_Toc166195188)

[2.4. Giao diện trang lưu trữ công văn 26](#_Toc166195189)

[2.5. Giao diện trang chỉnh sửa công văn 28](#_Toc166195190)

[3. Kết quả kiểm tra 28](#_Toc166195191)

[PHẦN KẾT LUẬN 29](#_Toc166195192)

[1. Kết quả đạt được 29](#_Toc166195193)

[2. Hướng phát triển 29](#_Toc166195194)

[TÀI LIỆU KHAM KHẢO 30](#_Toc166195195)

**DANH MỤC HÌNH**

[Hình 1: Django 11](#_Toc166196934)

[Hình 2: MongoDB 12](#_Toc166196935)

[Hình 3: Quá trình xử lý của công nghệ OCR 13](#_Toc166196936)

[Hình 4: Thư viện PIL 15](#_Toc166196937)

[Hình 5: Biểu đồ độ chính xác giữa hai mô hình OCR 22](#_Toc166196938)

[Hình 6: Giao diện trang chủ 24](#_Toc166196939)

[Hình 7: Giao diện trang quản lý 25](#_Toc166196940)

[Hình 8: Giao diện trang tìm kiếm 26](#_Toc166196941)

[Hình 9: Giao diện trang Upload (nhận dạng) 27](#_Toc166196942)

[Hình 10: Giao diện trang Upload (Hiển thị) 27](#_Toc166196943)

[Hình 11: Giao diện trang chỉnh sửa 28](#_Toc166196944)

**DANH MỤC BẢNG**

[Bảng 1: Collection "document" 18](#_Toc166197680)

**ABSTRACT**

Today, technology is increasingly developing and the country is entering a period of national digital transformation. There are many challenges that have occurred during this transformation process. Especially in the process of managing work and information. Therefore, optimizing the management of official documents by storing them on information systems becomes very important. This topic focuses on researching technologies for practical application to effectively solve this process.

Documents managed in the traditional way often require a lot of time and effort, especially storing and searching for information related to documents. This can cause loss of important information and confidential information may be exposed to the outside world. In addition, answering citizens' questions requires agility and accuracy, which is difficult to meet with traditional methods.

This topic offers a solution to help manage dispatches neatly and quickly, but also contributes to promoting the rapid national digital transformation process. Currently, artificial intelligence is used a lot in automation processes, reducing manual work and being much faster than traditional methods. In addition, this topic not only helps organizations and businesses manage documents effectively but also keeps up with the country's digitalization trend, opening up many opportunities to interact with the world. The model used in this topic is Tesseract developed by Google, tested on 10 files (including 10 image files and 10 original files for comparison) and the results showed a high accuracy of up to 81. 38%.

**TÓM TẮT**

Ngày nay, công nghệ ngày càng phát triển đất nước đã và đang bước vào thời kỳ chuyển đổi số quốc gia, có nhiều thách thức trong quá trình chuyển đổi này đã xảy ra. Đặc biệt là trong quá trình quản lý công việc và thông tin. Vì vậy, việc tối ưu hóa quản lý các công văn bằng cách lưu trữ chúng trên những hệ thống thông tin trở nên rất quan trọng. Đề tài này tập trung nghiên cứu những công nghệ nhằm ứng dụng vào thực tế để giải quyết hiệu quả quá trình này.

Công văn được quản lý theo cách truyền thống thường đòi hỏi nhiều thời gian và công sức, đặc biệt là lưu trữ và tìm kiếm thông tin có liên quan đến công văn. Việc này có thể gây mất mát thông tin quan trọng và những thông tin mật có thể bị lộ ra bên ngoài. Ngoài ra việc trả lời các thắc mắc của công dân đòi hỏi sự nhanh nhạy và chính xác, đây là điều mà các phương pháp truyền thống khó có thể đáp ứng.

Đề tài này đưa ra giải pháp giúp quản lý công văn một cách gọn gàng, nhanh chóng mà còn góp phần giúp thúc đẩy quá trình chuyển đổi số quốc gia được diễn ra nhanh chóng. Hiện nay, trí tuệ nhân tạo được sử dụng rất nhiều trong các quá trình tự động hóa, giảm các công việc phải làm thủ công và nhanh hơn rất nhiều so với các phương pháp truyền thống. Ngoài ra, đề tài này không chỉ giúp các tổ chức, doanh nghiệp quản lý công văn hiệu quả mà còn đuổi kịp với xu thế số hóa của quốc gia, mở ra nhiều cơ hội hội nhạp với thế giới. Mô hình được sử dụng ở đề tài này là Tesseract do Google phát triển, được kiểm tra trên 10 file (bao gồm 10 file hình ảnh và 10 file gốc để so sánh) kết quả cho ra độ chính xác khác cao lên đến 81,38%.

# PHẦN GIỚI THIỆU

## Đặt vấn đề.

Trong những năm gần đây với sự phát triển không ngừng của khoa học kỹ thuật, đặt biệt là lĩnh vực công nghệ thông tin với các ứng dụng của công nghệ thông tin vào các lĩnh vực khác, đã góp phần to lớn cho sự phát triển của nhân loại. Trong các lĩnh vực đó, lĩnh vực quản lí thực sự giúp ích rất nhiều cho con người. Thực hiện chuyển đổi số quốc gia trong lĩnh vực quản lý là một trong những hoạt động được nhiều sự quan tâm và ưu tiên của của các quốc gia. Trên cơ sở Quyết định số 131/QĐ-TTg năm 2022 của Thủ tướng Chính phủ phê duyệt đề án “Tăng cường ứng dụng công nghệ thông tin và chuyển đổi số trong giai đoạn 2022 - 2025, định hướng đến năm 2030”, trường Đại học Cần Thơ đã có những hành động để thích ứng với bối cảnh hiện nay.

Một số tổ chức thực hiện chuyển đổi số cần trải qua giai đoạn đầu tiên là “số hóa”. Đây là quá trình chuyển đổi các thực thể từ vật lý sang dạng số để có thể lưu trữ trên máy tính. Ví dụ như những thông tin cơ bản của một công dân có thể lưu trữ thành các trường trên cơ sở dữ liệu. Các số liệu kinh doanh từ hóa đơn, chứng từ,… khi cần có thể được tổ chức thành các bảng số liệu lưu vào Microsoft Excel. Hoặc chỉ lưu trữ những tệp tin vào máy tính chứa các dữ liệu cũng có thể gọi là số hóa.

Trong thực tế vẫn còn nhiều dữ liệu đang được lưu hành dưới dạng vật lý, như các công văn, quyết định,… Khi cần tra cứu, tham khảo thì phải đến tận nơi để tìm và xem. Đây là một thử thách lớn trong quá trình chuyển đổi số nói chung.

Vấn đề này còn đặt ra những câu hỏi để đảm bảo chất lượng nhận diện OCR để giảm thiểu lỗi và tăng cường độ chính xác của thông tin. Cách tạo ra một giao diện đơn giản thân thiện với người dùng, đảm bảo việc quản lý công văn trở nên đơn giản và chính xác cũng là một điểm quan trọng.

Ngoài ra, vấn đề bảo mật thông tin cũng là một khía cạnh đáng quan tâm. Cần phải có những biện pháp bảo mật an toàn để đảm bảo rằng thông tin lưu trữ không bị lộ ra ngoài làm ảnh hưởng đến doanh nghiệp, đơn vị hay quốc gia.

1. **Lịch sử giải quyết vấn đề.**

Giải pháp cho các vấn đề nêu trên được đề xuất, nghiên cứu và đánh giá các, công nghệ hiện đại liên quan đến số hóa tài liệu, văn bản. Cụ thể như sau:

Công nghệ nhận dạng ký tự quang học là một phương pháp nhận dạng văn bản hoặc ký tự bằng cách sử dụng các phép đo và phân tích các thuộc tính quang học của chữ viết tay hoặc in ấn. Phương pháp này dựa trên việc quan sát và phân tích các biến đổi trong ánh sáng khi chiếu qua hoặc phản xạ từ các ký tự. Công nghệ này hoạt động bằng cách sử dụng các cảm biến quang học để thu thập dữ liệu từ mẫu văn bản hoặc ký tự. Sau đó, các thuật toán phân tích dữ liệu để xác định và nhận dạng các ký tự cụ thể. Một trong những phương pháp phổ biến nhất là sử dụng máy ảnh hoặc cảm biến quang học để chụp ảnh văn bản hoặc ký tự, sau đó sử dụng các thuật toán nhận dạng hình ảnh để nhận dạng và giải mã chúng. Dữ liệu quang học được thu thập từ mẫu văn bản bao gồm các thuộc tính như độ sáng, độ tương phản, đặc điểm hình dáng và kết cấu. Các thuật toán phân tích dữ liệu này để xác định các ký tự và từ đó giải mã văn bản. Công nghệ nhận dạng ký tự quang học được sử dụng rộng rãi trong các ứng dụng như quét mã vạch, nhận dạng ký tự trong các tài liệu in, nhận diện chữ viết tay trong bức thư hoặc ghi chú, và trong các hệ thống giao dịch tài chính. Một số thách thức của công nghệ này bao gồm khả năng nhận dạng chính xác trong điều kiện ánh sáng yếu, đa dạng về kiểu chữ và kết cấu văn bản, cũng như khả năng xử lý nhanh chóng trong thời gian thực.

1. **Mục tiêu đề tài.**

Mục tiêu đề tài “Xây dựng website quản lý công văn trường CNTT & TT tích hợp nhận dạng văn bản” là tạo ra một hệ thống giúp tổ chức quản lý các công văn một cách hiệu quả. Hệ thống sử dụng công nghệ nhận dạng ký tự quan học OCR để để chuyển đổi thông tin từ hình ảnh và tài liệu quét sang dạng văn bản có thể chỉnh sửa và tìm kiếm những từ khóa được lưu trữ trong cơ sở dữ liệu.

Tích hợp nhận diện ký tự quang học để tự động chuyển đổi các tài liệu hình ảnh hoặc PDF thành dạng văn bản có thể tìm kiếm. Giúp người dùng tìm kiếm và truy cập dự trên các nội dung công văn của chúng thay vì chỉ dựa vào tiêu đề.

Phát triển giao diện người dùng dễ sử dụng và có thể tìm kiếm và quản lý các công văn một cách hiệu quả.

1. **Đối tượng và phạm vi nghiên cứu của đề tài.**

Đối tượng nghiên cứu của đề tài là bao gồm cả website quản lý công văn và người sử dụng hệ thống này. Đối chính vẫn là các tài liệu công văn.

Hệ thống xây dựng trên các công nghệ hiện tại với mục tiêu cung cấp giải pháp để quản lý công văn và nhận dạng ký tự quang học hiệu quả.

Phạm vi nghiên cứu của đề tài bao gồm việc phân tích yêu cầu, thiết kế, xây dựng và kiểm thử hệ thống.

Giao diện thân thiện dễ sử dụng và có thể học các sử dụng một cách nhanh chóng.

Chức năng nhận dạng ký tự quang học (OCR) sẽ được tích hợp vào hệ thống.

1. **Nội dung nghiên cứu.**

* Quy trình nghiên cứu: cách tổ chức lưu trữ dữ liệu của cơ sở dữ liệu MongoDB, cách thức hoạt động của Django, cách nhận dạng ký tự của mô hình Google Tesseract.
* Công nghệ sử dụng: Django, MongoDB, Google Tesseract,…
* Công cụ hỗ trợ xây dựng và phát triển: Visual Studio Code, MongoDB Compass.

1. **Kết quả đạt được.**

Phát triển được hệ thống quản lý công văn, khả năng tổ chức lưu trữ tập tin một cách dễ dàng nhanh chóng. Điều này giúp giảm bớt sự phức tạp trong việc quản lý công văn.

Tích hợp thành công chức năng nhận diện văn bản vào website. Nhờ chức năng này người dùng có thể tải tài liệu lên và quét các tài liệu để có thể tự động điền vào các trường trong form nhập liệu. Giúp nâng cao hiệu suất trong công việc.

1. **Bố cục báo cáo.**

**Phần giới thiệu**

Giới thiệu tổng quát về đề tài.

**Phần nội dung**

**Chương 1:** Mô tả bài toán.

**Chương 2:** Thiết kế và cài đặt.

**Chương 3:** Kiểm thử và đánh giá.

**Phần kết luận**

Trình bày kết quả đạt được và hướng phát triển hệ thống.

# PHẦN NỘI DUNG

# CHƯƠNG 1: MÔ TẢ BÀI TOÁN

1. **Mô tả chi tiết bài toán**

Xây dựng website để quản lý công văn có tích hợp nhận dạng ký tự quang học là bài toán sử dụng giải thuật liên quan đến OCR (Optical Character Recognition) - là một công nghệ sử dụng để nhận dạng và trích xuất văn bản từ hình ảnh hoặc cái tài liệu điện tử như ảnh chụp tài liệu, tập tin PDF hoặc scan từ các tài liệu giấy.

Mục tiêu là xây dựng một website có khả năng quản lý công văn một cách hiệu quả và tiện lợi cho việc tra cứu. Website cho phép người dùng dễ dàng tiềm kiếm, xem, lưu trữ và tra cứu các công văn.

Và đồng thời website còn cần có khả năng nhận diện được ký tự quang học từ các công văn đã được quét, chụp hoặc từ các file tài liệu. Nó cho phép việc tự động nhập dữ liệu khi lưu trữ giúp tiết kiệm thời gian của người dùng. Chính vì vậy công nghệ OCR cần được tích hợp vào website.

Hệ thống website sẽ được thiết kế như hình …, có thể thấy quy trình hoạt động của hệ thống như sau:

* Người dùng tiến hành up load file PDF lên hệ thống và tiến hành nhận dạng
* Sau đó file PDF của người dùng sẽ được hệ thống chuyển đổi thành hình ảnh và tiến hành nhận dạng ký tự
* Sau khi nhận dạng hệ thống sẽ bắt đầu rút trích những thông tin đặc trưng và điền vào form hiển thị cho người dùng
* Trong trường hợp còn sai lỗi chính tả hoặc các thông tin thì người dùng có thể hiệu chỉnh bằng tay và lưu vào hệ thống.

1. **Vấn đề và giải pháp liên quan đến bài toán**
   1. **Xây dựng và huấn luyện mô hình** 
      1. **Thu thập dữ liệu và tiền xử lý dữ liệu**

Dữ liệu là các loại tài liệu như thông báo, nghị quyết, quyết định, nghị định được thu thập từ các trang web trên Internet như chinhphu.vn, vnsw.gov.vn các loại tài liệu này được thu thập dưới dạng file PDF. Sau khi thu thập tiến hành chọn lọc những tài liệu rõ, không bị mờ.

* + 1. **Huấn luyện mô hình**

Sử dụng mô hình Google Tesseract để nhận dạng văn bản. Tesseract là một phần mềm mã nguồn mở được phát triển bởi Google, được dùng cho việc nhận dạng ký tự (OCR). Dựa trên một kiến trúc máy đa lớp, Tesseract có khả năng nhận dạng được văn bản từ hình ảnh với mức độ chính xác khá cao.

Tesseract đi kèm với các mô hình đã được huấn luyện cho nhiều ngôn ngữ, bao gồm cả Tiếng Việt nên không yêu cầu quá trình huấn luyện phức tạp.

1. **Mô tả giải pháp cho bài toán**
   1. **Django**

****

Hình 1: Django

Django là một framework phát triển web mã nguồn mở, được viết bằng ngôn ngữ lập trình Python. Nó sử dụng mô hình MTV (Model-Template-View) để tổ chức mã nguồn và logic của ứng dụng web. Trong mô hình này, Model đại diện cho cơ sở dữ liệu và quản lý truy cập vào dữ liệu, Template là phần giao diện người dùng, và View là nơi xử lý logic của ứng dụng.

Một trong những điểm mạnh của Django là ORM (Object-Relational Mapping), cho phép tương tác với cơ sở dữ liệu thông qua các đối tượng Python, giúp giảm thiểu sự phụ thuộc vào SQL truyền thống. Nó cung cấp tính năng bảo mật tích hợp như xác thực người dùng, phân quyền và bảo vệ chống CSRF (Cross-Site Request Forgery).

Django cũng đi kèm với nhiều tính năng tích hợp sẵn như quản lý biểu mẫu, xử lý yêu cầu và phản hồi HTTP, xử lý hình ảnh và email. Nó cung cấp một cách tiếp cận linh hoạt và dễ dàng mở rộng cho việc phát triển các ứng dụng web phức tạp.

Với cộng đồng lớn và tài liệu phong phú, việc học và sử dụng Django trở nên dễ dàng và thuận tiện hơn bao giờ hết. Đây là lý do mà Django đã trở thành một trong những framework phát triển web phổ biến và được ưa chuộng trong cộng đồng phát triển phần mềm.

* 1. **MongoDB**

****

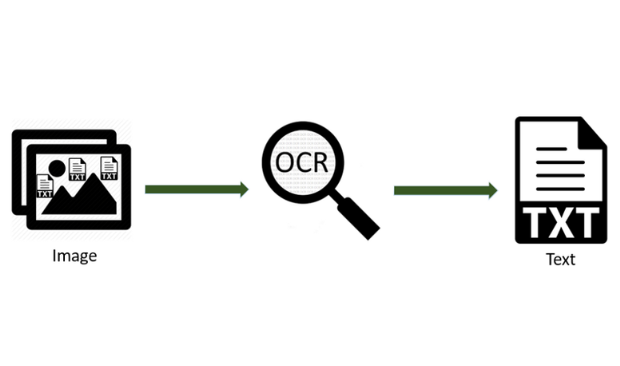
Hình 2: MongoDB

MongoDB là hệ CSDL mã nguồn mở, là CSDL phi quan hệ hay còn gọi là NoSQL (None-Relationship SQL hay còn gọi là Not only SQL). NoSQL được phát triển trên Javascript Framework với kiểu dữ liệu là JSON và dạng dữ liệu theo kiểu key và value. NoSQL ra đời như là sự bổ sung cho những khuyết điểm và thiếu xót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS (Relational Database Management System - Hệ quản trị cơ sở dữ liệu quan hệ) về tốc độ, tính năng, khả năng mở rộng.

Với NoSQL ta có thể mở rộng dữ liệu mà không lo tới những việc như tạo khóa ngoại, khóa chính, kiểm tra ràng buộc. NoSQL bỏ qua tính toàn vẹn của dữ liệu và transaction để đổi lấy hiệu suất nhanh và khả năng mở rộng. MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.

Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là collection thay vì bảng. So với RDBMS thì trong MongoDB collection ứng với table, còn document sẽ ứng với row, MongoDB sẽ dùng các document thay cho row trong RDBMS.

* 1. **Phương pháp nhận dạng văn bản**
     1. **Khái niệm về OCR (Optical Character Recognition)**

****

Hình 3: Quá trình xử lý của công nghệ OCR

OCR (Optical Character Recognition) là một công nghệ được sử dụng để nhận dạng và trích xuất văn bản từ hình ảnh hoặc các tập tin điện tử. Công nghệ này cho phép máy tính "đọc" và hiểu được các ký tự được hiển thị trong các hình ảnh, tài liệu quét, hoặc các file PDF.

Cơ chế hoạt động của OCR thường bao gồm các bước sau:

* Tiền xử lý hình ảnh: Ảnh hoặc tài liệu quét thường cần được tiền xử lý để loại bỏ nhiễu, cải thiện chất lượng hình ảnh, và chuẩn bị cho quá trình nhận dạng ký tự.
* Phân đoạn hình ảnh: Hình ảnh được chia nhỏ thành các phần nhỏ hơn, mỗi phần chứa một ký tự hoặc một phần của một ký tự.
* Nhận dạng ký tự: Các phần của hình ảnh được phân đoạn được phân tích để xác định các ký tự hoặc ký tự có thể. Các thuật toán và mô hình máy học được sử dụng để so khớp các hình ảnh với các ký tự đã biết.
* Hậu xử lý: Sau khi các ký tự được nhận dạng, các bước hậu xử lý có thể được thực hiện để cải thiện chính xác của văn bản được trích xuất, bao gồm việc sửa lỗi chính tả, phục hồi từ các lỗi nhận dạng, hoặc kết hợp các ký tự lại với nhau để tạo thành từ hoặc câu.

OCR có nhiều ứng dụng trong thực tế, bao gồm chuyển đổi tài liệu giấy thành dạng điện tử, trích xuất dữ liệu từ hình ảnh hoặc tài liệu quét để nhập vào các hệ thống thông tin, và hỗ trợ trong việc tìm kiếm và phân loại thông tin từ văn bản số.

* + 1. **Một số thư viện nhận diện ký tự quang học**
       1. **EasyOCR**

EasyOCR là một thư viện mã nguồn mở được thiết kế để giúp nhận dạng ký tự từ hình ảnh hoặc các tập tin PDF một cách dễ dàng và nhanh chóng. Với khả năng sử dụng trên ngôn ngữ lập trình Python và giao diện đơn giản, EasyOCR trở thành một công cụ hữu ích cho việc tích hợp vào các dự án và ứng dụng của các nhà phát triển.

Đặc biệt, EasyOCR hỗ trợ nhận dạng văn bản cho nhiều ngôn ngữ khác nhau trên thế giới, từ các ngôn ngữ phổ biến đến các ngôn ngữ nguồn gốc phức tạp. Khả năng linh hoạt của nó cho phép nhận dạng văn bản từ nhiều định dạng hình ảnh phổ biến như JPEG, PNG và TIFF.

Với EasyOCR, việc tích hợp vào ứng dụng Python trở nên đơn giản chỉ với một vài dòng mã. Điều này giúp cho việc nhận dạng văn bản từ hình ảnh trở nên dễ dàng và thuận tiện, phù hợp cho các ứng dụng như xử lý tài liệu, phát hiện và trích xuất thông tin từ hình ảnh, và nhiều ứng dụng khác trong thực tế.

* + - 1. **TesseractOCR**

TesseractOCR là một công cụ mã nguồn mở mạnh mẽ được phát triển bởi Google, chuyên dùng để nhận dạng ký tự từ hình ảnh hoặc các tập tin PDF. Sử dụng mạng nơ-ron, TesseractOCR cung cấp khả năng nhận dạng văn bản với độ chính xác cao trên nhiều ngôn ngữ khác nhau trên thế giới.

Điểm đặc biệt của TesseractOCR là tính đa ngôn ngữ, cho phép nhận dạng văn bản từ một loạt các ngôn ngữ, từ các ngôn ngữ phổ biến đến các ngôn ngữ nguồn gốc phức tạp. Với giao diện dòng lệnh hoặc các giao diện lập trình ứng dụng (API), việc sử dụng TesseractOCR trở nên dễ dàng và thuận tiện cho các nhà phát triển.

Ngoài ra, TesseractOCR cũng được ưa chuộng vì tính mở rộng và sự hỗ trợ từ cộng đồng lớn. Điều này giúp cải thiện và mở rộng tính năng của công cụ liên tục.

TesseractOCR đã được áp dụng rộng rãi trong nhiều lĩnh vực thực tế như chuyển đổi tài liệu giấy thành dạng điện tử, trích xuất thông tin từ hình ảnh, và xử lý văn bản trong các hệ thống quản lý tài liệu. Với khả năng mạnh mẽ và linh hoạt của mình, TesseractOCR là một lựa chọn ưu việt cho những ai cần một giải pháp nhận dạng văn bản từ hình ảnh.

* + 1. **Thư viện PIL**

****

Hình 4: Thư viện PIL

Thư viện PIL (Python Imaging Library) là một công cụ mạnh mẽ và linh hoạt cho việc xử lý hình ảnh trong Python. Được phát triển bởi Fredrik Lundh và nhóm các nhà phát triển, PIL cung cấp một loạt các chức năng để mở, chỉnh sửa, và lưu trữ hình ảnh theo nhiều định dạng khác nhau.

Với PIL, việc xử lý hình ảnh trở nên đơn giản và hiệu quả. Có thể mở và lưu trữ hình ảnh dễ dàng, cũng như thực hiện các thao tác chỉnh sửa như điều chỉnh độ sáng, tương phản, độ mờ và màu sắc. Thêm vào đó, PIL cung cấp các công cụ để thay đổi kích thước, cắt, quay và lật hình ảnh theo nhu cầu của bạn.

Thư viện này hỗ trợ một loạt các định dạng hình ảnh phổ biến như JPEG, PNG, GIF, BMP, và TIFF. Ngoài ra, PIL cũng cho phép bạn đọc và ghi dữ liệu Exif, giúp bạn truy cập thông tin về camera và cài đặt khi chụp hình.

Tích hợp PIL vào các dự án Python cũng trở nên đơn giản và linh hoạt. Tính linh hoạt và khả năng mở rộng của PIL đã làm cho nó trở thành một công cụ ưa chuộng trong cộng đồng phát triển Python, và được sử dụng rộng rãi trong nhiều lĩnh vực như xử lý ảnh y khoa, xử lý ảnh số, và các ứng dụng liên quan đến đồ họa máy tính.

* + 1. **Thư viện pdf2img**

Thư viện pdf2img là một công cụ quan trọng và hữu ích trong Python cho việc chuyển đổi các tập tin PDF thành các hình ảnh. Bằng cách sử dụng pdf2img, người dùng có thể dễ dàng trích xuất nội dung từ các tập tin PDF và chuyển đổi chúng thành các hình ảnh như JPEG hoặc PNG, để tiện lợi sử dụng trong các ứng dụng và dự án khác nhau.

Với pdf2img, việc chuyển đổi từng trang của tập tin PDF thành các hình ảnh trở nên đơn giản và thuận tiện. Người dùng cũng có thể tuỳ chỉnh định dạng và chất lượng của hình ảnh đầu ra, đảm bảo rằng chúng phù hợp với yêu cầu cụ thể của ứng dụng hoặc dự án.

Giao diện dễ sử dụng của pdf2img giúp người dùng thực hiện quá trình chuyển đổi một cách nhanh chóng và hiệu quả chỉ với vài dòng mã. Công cụ này rất hữu ích cho các ứng dụng như trình chiếu, website hoặc báo cáo, nơi mà việc sử dụng hình ảnh từ các tập tin PDF là cần thiết để truyền đạt thông tin một cách rõ ràng và sinh động.

# CHƯƠNG 2: THIẾT KẾ VÀ CÀI ĐẶT

1. **Thiết kế hệ thống** 
   1. **Các chức năng của website**

Website bao gồm các chức năng giúp cho người dùng thuận tiện trong việc lưu trữ, tra cứu và tìm kiếm công văn. Các chức có trong website như:

* Giao diện người dùng: Trang chủ bao gồm thanh silde bar chứ các chức năng của trang web, một ô tìm kiếm. song nó cũng chứa những thông tin về các sự kiện.
* Chức năng tìm kiếm: người dùng có thể nhập bất kì từ khóa nào và nhấn nút tìm kiếm để tìm thông tin mong muốn.
* Chức năng tải file PDF lên để nhận dạng ký tự quang học: người chọn file pdf để tải lên và sử hoặc điền các thông tin còn thiếu vào form để tiến hành lưu file.
* Quản lý công văn: người dùng có thể chỉnh sửa chi tiết các công văn, xóa hoặc tải file công văn về máy.

**1.2. Cơ sở dữ liệu**

MongoEngine là một công cụ ODM (Object-Document Mapper) cho MongoDB trong Python. Nó cung cấp một cách tiện lợi để làm việc với MongoDB bằng cách ánh xạ các đối tượng Python sang các tài liệu MongoDB và ngược lại.

Cụ thể, MongoEngine cho phép định nghĩa các lớp đối tượng Python, mỗi lớp tương ứng với một collection trong MongoDB. Bằng cách sử dụng các trường (fields) được cung cấp bởi MongoEngine, có thể mô tả cấu trúc của các tài liệu trong collection. MongoEngine cũng cung cấp các phương thức để thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên các đối tượng Python này, và tự động ánh xạ chúng vào cơ sở dữ liệu MongoDB.

MongoEngine giúp đơn giản hóa việc tương tác với MongoDB từ Python, cung cấp một giao diện dễ sử dụng và linh hoạt cho việc lập trình ứng dụng dựa trên MongoDB. Nó phù hợp cho các dự án Python muốn sử dụng MongoDB làm cơ sở dữ liệu.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Type** | **Primary Key** | **Foreign Key** | **Mô tả** |
| \_id | ObjectId | **x** |  | ID công văn |
| organization\_name | StringField |  |  | Tên cơ quan |
| email | StringField |  |  | Email |
| field | StringField |  |  | Lĩnh vực |
| number | StringField |  |  | Số công văn |
| pdf\_file | FileField |  |  | Tệp tin PDF |
| issuance\_date | DateTimeField |  |  | Ngày ban hành |

Bảng 1: Collection "document"

1. **Thiết kế và cài đặt giải thuật**

Trong dự án này áp dụng công nghệ TesseractOCR để nhận dạng ký tự quang học vì Teeseract là một phần mềm mã nguồn mở do Google phát triển và được cộng đồng quốc tế hỗ trợ. Nó có ưu điểm là hỗ trợ nhiều ngôn ngữ trên thế giới trong đó có cả Tiếng Việt mà không cần trải qua quá trình huấn luyện.

* 1. **Nhận diện số công văn**

Nhận vào một chuỗi văn bản text mà TesseractOCR đã nhận dạng được và thực hiện việc trích xuất số công văn từ trong đó. Để thực hiện điều này, sử dụng biểu thức chính quy (regular expression) để tìm kiếm mẫu có dạng "*Số: <số>/<chữ cái>*" trong văn bản.

Biểu thức chính quy được sử dụng là *r ' Số:\s\*(\d+\ /[A-Z]+)'*, trong đó:

* Số: là chuỗi ký tự cần phải xuất hiện trước số công văn.
* \s\* là một hoặc nhiều ký tự trống (bao gồm cả khoảng trắng, tab, xuống dòng) để cho phép có một số khoảng trắng trước số công văn.
* *(\d+\/[A-Z]+)* là một nhóm ký tự bao gồm:
  + *\d+* đại diện cho một hoặc nhiều chữ số.
  + *\ /* đại diện cho dấu gạch chéo (/) phải xuất hiện ngay sau số.
  + *[A-Z]+* đại diện cho một hoặc nhiều chữ cái viết hoa.

Sau đó, hàm sử dụng phương thức re.search() để tìm kiếm mẫu trong văn bản. Nếu tìm thấy, hàm trả về phần tử đầu tiên của kết quả tìm kiếm, tức là số công văn. Nếu không tìm thấy sẽ trả về “*None*”.

* 1. **Nhận diện ngày ban hành công văn**

Nhận vào một chuỗi văn bản TesseractOCR đã nhận dạng được và thực hiện việc trích xuất thông tin về ngày tháng từ trong đó. Để thực hiện điều này, sử dụng biểu thức chính quy (regular expression) để tìm kiếm mẫu có dạng "*Tên địa danh ngày xx tháng xx năm xxxx*" trong văn bản.

Biểu thức chính quy được sử dụng là *r"(\D+) ngày (\d{1,2}) tháng (\d{1,2}) năm (\d{4})"*, trong đó:

* *(\D+)* là một nhóm ký tự không phải là số để đại diện cho tên địa danh.
* *‘ngày’* là chuỗi ký tự cần phải xuất hiện trước ngày.
* *(\d{1,2})* là một nhóm ký tự đại diện cho ngày, có thể là một hoặc hai chữ số.
* ‘tháng’ là chuỗi ký tự cần phải xuất hiện trước tháng.
* *(\d{1,2})* là một nhóm ký tự đại diện cho tháng, có thể là một hoặc hai chữ số.
* *‘năm’* là chuỗi ký tự cần phải xuất hiện trước năm.
* *(\d{4})* là một nhóm ký tự đại diện cho năm, bao gồm đúng bốn chữ số.

Sau đó, hàm sử dụng phương thức *re.findall()* để tìm tất cả các mẫu trong văn bản. Nếu tìm thấy, hàm sẽ trích xuất thông tin về tên địa danh, ngày, tháng và năm từ kết quả tìm kiếm. Tiếp theo, hàm kiểm tra tính hợp lệ của ngày tháng năm bằng cách sử dụng *datetime*, và trả về một chuỗi ngày tháng đã được định dạng đúng (theo định dạng *'dd/mm/yyyy'*) cùng với tên địa danh. Nếu không tìm thấy kết quả, hoặc ngày tháng năm không hợp lệ, hàm sẽ trả về *“None”.*

* 1. **Nhận dạng email**

Nhận vào một chuỗi văn bản TesseractOCR đã nhận dạng được và thực hiện việc trích xuất địa chỉ email từ trong đó. Để thực hiện điều này, sử dụng biểu thức chính quy (regular expression) để tìm kiếm chuỗi có dạng email trong văn bản.

Biểu thức chính quy được sử dụng là *r'\b[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'*, trong đó:

* *\b* đại diện cho biên từ (word boundary), đảm bảo rằng email được tìm kiếm không phải là một phần của một từ khác.
* *[A-Za-z0-9.\_%+-]+* đại diện cho tên người dùng (username) trong email, bao gồm các ký tự chữ cái, số, và các ký tự đặc biệt như dấu chấm (.), gạch dưới (\_), phần trăm (%), dấu cộng (+), và dấu gạch ngang (-).
* *@ là ký tự "@".*
* *[A-Za-z0-9.-]+* đại diện cho tên miền (domain) trong email, cũng bao gồm các ký tự chữ cái, số, dấu chấm (.) và dấu gạch ngang (-).
* *\.* là ký tự dấu chấm (.) được escape bằng dấu gạch chéo ().
* *[A-Z|a-z]{2,}* đại diện cho phần đuôi (suffix) của tên miền, có thể là các ký tự chữ cái và có ít nhất hai ký tự.
* *\b* đại diện cho biên từ (word boundary) ở cuối.

Sau khi tìm kiếm, nếu tìm thấy một chuỗi có dạng email sẽ trả về email đó. Nếu không tìm thấy, hàm sẽ trả về *“None”*.

* 1. **Nhận dạng tên cơ quan ban hành**

Nhận vào một chuỗi văn bản TesseractOCR đã nhận dạng được và thực hiện việc trích xuất tên tổ chức từ trong đó. Để thực hiện điều này, hàm sử dụng biểu thức chính quy (regular expression) để tìm kiếm từ khóa "Cơ quan:" trong văn bản.

Biểu thức chính quy được sử dụng là *r'Cơ quan:\s\*(.\*)',* trong đó:

* *“Cơ quan”* là chuỗi ký tự cần phải xuất hiện trước tên tổ chức.
* *\s\** là một hoặc nhiều ký tự trống (bao gồm cả khoảng trắng, tab, xuống dòng) để cho phép có một số khoảng trắng trước tên tổ chức.
* *(.\*)* là một nhóm ký tự đại diện cho tên tổ chức, có thể là bất kỳ ký tự nào (bao gồm cả khoảng trắng) và có thể xuất hiện nhiều lần.

Sau khi tìm kiếm, nếu tìm thấy từ khóa "*Cơ quan:*", hàm sẽ trả về phần trùng khớp của biểu thức chính quy, tức là phần sau từ khóa "*Cơ quan*:". Nếu không tìm thấy trả về *“None”.*

* 1. **Phân loại lĩnh vực công văn**

Nhận vào một chuỗi văn bản TesseractOCR đã nhận dạng được và thực hiện việc phân loại nó vào một trong các lĩnh vực khác nhau dựa trên từ khóa xuất hiện trong văn bản. Để thực hiện điều này, hàm sử dụng một từ điển keywords chứa các từ khóa tương ứng với mỗi lĩnh vực.

Cụ thể, từ điển keywords bao gồm các lĩnh vực như "Khoa học", "Kinh tế", "Y tế", "Công nghệ", và "Chính trị", cùng với danh sách các từ khóa liên quan đến mỗi lĩnh vực. Các từ khóa cụ thể như sau:

* Khoa học (*'khoa học', 'nghiên cứu', 'kỹ thuật', 'công nghệ'*)
* Kinh tế (*'kinh tế', 'tài chính', 'doanh nghiệp', 'thị trường'*)
* Y tế (*'y tế', 'bệnh', 'bác sĩ', 'bệnh viện'*)
* Công nghệ (*'công nghệ', 'thiết bị', 'ứng dụng', 'máy tính'*)
* Chính trị (*'chính trị', 'chính phủ', 'bầu cử', 'quốc hội'*)

Sau đó, hàm đếm số lần xuất hiện của từ khóa cho mỗi lĩnh vực trong văn bản. Sau khi đếm xong, hàm tìm lĩnh vực có số lần xuất hiện của từ khóa cao nhất và trả về lĩnh vực đó. Nếu không có từ khóa nào được tìm thấy trong văn bản sẽ trả về lĩnh vực mặc định là "*khác*".

1. **Đánh giá mô hình** 
   1. **Đánh giá độ chính xác của từng mô hình**

Dùng thư viện *difflib* để đánh giá độ chính xác của kết quả. Tạo đối tượng *SequenceMatcher*: Đầu tiên, bạn cần tạo một đối tượng *SequenceMatcher* bằng cách cung cấp hai chuỗi cần so sánh cho nó. Đối tượng này sẽ phân tích hai chuỗi và tìm ra các phần giống và phần khác nhau giữa chúng.

Tìm ra các đoạn tương đồng và khác biệt: Sau khi đã tạo đối tượng bằng *SequenceMatcher*, có thể sử dụng các phương thức như *get\_opcodes()* để tìm ra các phần tương đồng và khác biệt giữa hai chuỗi. Các phần tương đồng sẽ được xác định dựa trên các dãy liên tục của các ký tự giống nhau trong hai chuỗi, trong khi các phần khác biệt sẽ là những phần không giống nhau.

Ví dụ, có chuỗi 1 là "Hello world" và chuỗi 2 là "Hello there", và ta muốn biết bao nhiêu phần trăm các ký tự trong chuỗi 1 cũng xuất hiện trong chuỗi 2, ta có thể sử dụng độ chính xác để làm điều này.

Đếm số lượng ký tự đúng: Số lượng ký tự trong chuỗi 1 mà cũng xuất hiện trong chuỗi 2. Trong ví dụ của chúng ta, các ký tự "H", "e", "l", "l", "o" từ chuỗi 1 cũng xuất hiện trong chuỗi 2. Đây là tổng số lượng ký tự trong chuỗi 1. Trong trường hợp của chúng ta, tổng số ký tự trong chuỗi 1 là 11. Để tính độ chính xác, chúng ta chỉ cần chia số lượng ký tự đúng cho tổng số ký tự trong chuỗi 1, và nhân kết quả với 100 để có phần trăm.

Thử nghiệm 2 mô hình để nhận dạng văn bản là EasyOCR và TesseractOCR với tập dữ liệu bao gồm 10 hình ảnh và 10 file chứa nội dung của hình ảnh nhận dạng để so sánh độ chính xác sau khi hai mô hình nhận diện được văn bản. Việc đánh giá độ chính xác được thực hiện bằng cách sau khi tiến hành nhận diện văn bản sẽ lưu vào file txt và sử dụng thư viện *difflib* trong python để so sánh và tìm kiếm sự khác biệt giữa các chuỗi văn bản. Kết quả độ chính xác trung bình sau 10 lần thử nghiệm lần lượt là EasyOCR (63,35%) và Tesseract (81,38%)

Hình 5: Biểu đồ độ chính xác giữa hai mô hình OCR

* 1. **So sánh độ hiệu quả của các mô hình**

Kết quả nhận diện của TesseractOCR đạt 81,38% độ chính xác này khá cao nhưng vẫn còn chưa đúng hoàn toàn, còn nhiều lỗi chính tả và xuất hiện những ký tự vô nghĩa,… Để tăng độ chính xác khi rút trích thông tin, ta nên tiến hành xử lý văn bản trước khi rút trích thông tin điền form.

Còn về kết quả nhận diện của EasyOCR đạt 63,35% với độ chính xác này cho thấy đây là một giải pháp kém hiệu quả hơn so với TesseractOCR do khác nhau đối với ngôn ngữ và kiểu chữ. Tốc độ xử lý của EasyOCR là lâu hơn. Và EasyOCR cung cấp ít tùy chọn tùy chỉnh hơn.

# CHƯƠNG 3: KIỂM THỬ VÀ ĐÁNH GIÁ

1. **Mục tiêu**

Mục tiêu kiểm thử của một trang web quản lý công văn có tích hợp nhận dạng văn bản bao gồm một loạt các bước để đảm bảo rằng hệ thống hoạt động một cách đáng tin cậy và hiệu quả.

Kiểm tra tính năng quản lý công văn:

* Xác thực và ủy quyền: Đảm bảo chỉ người dùng được phép có quyền truy cập và thực hiện các hoạt động quản lý công văn.
* Tạo, chỉnh sửa và xóa công văn: Kiểm tra tính đúng đắn và hoạt động của chức năng này.
* Tìm kiếm và lọc công văn: Đảm bảo tính chính xác và hiệu suất của việc tìm kiếm và lọc công văn dựa trên các tiêu chí khác nhau.

Kiểm tra tính năng nhận dạng văn bản:

* Độ chính xác của OCR: Kiểm tra độ chính xác của việc nhận dạng văn bản từ các hình ảnh hoặc tệp PDF.
* Xử lý văn bản đa ngôn ngữ: Kiểm tra khả năng nhận dạng văn bản.
* Xử lý hình ảnh: Đảm bảo hệ thống có thể xử lý các hình ảnh với chất lượng khác nhau một cách hiệu quả.

Kiểm tra tính tương thích và hiệu suất:

* Tương thích trình duyệt: Kiểm tra tính tương thích của trang web trên các trình duyệt phổ biến.
* Đáp ứng: Đảm bảo trang web hoạt động một cách mượt mà trên các thiết bị và kích thước màn hình khác nhau.
* Hiệu suất: Kiểm tra thời gian tải trang và đáp ứng của hệ thống dưới các tải trọng khác nhau.

Kiểm tra tính năng khác:

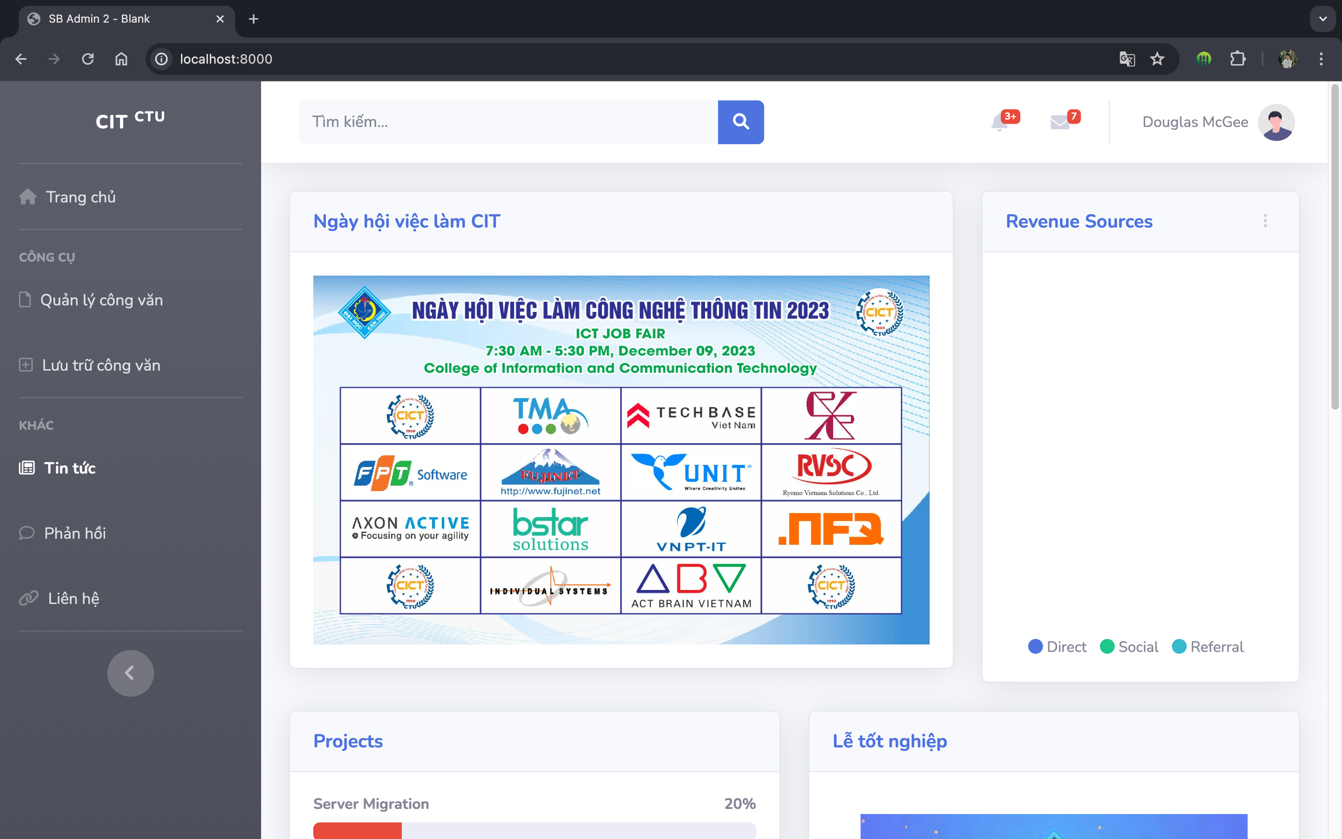
* Giao diện người dùng: Kiểm tra tính thẩm mỹ và dễ sử dụng của giao diện người dùng.
* Báo cáo và thống kê: Kiểm tra tính đúng đắn và tính khả dụng của báo cáo và thống kê được tạo ra từ dữ liệu quản lý công văn.

1. **Giao diện và chức năng**

Khi người dùng truy cập vào trang web, toàn bộ nội dung web sẽ được hiển thị. Website bao gồm các trang như: trang chủ, trang tìm kiếm các công văn đã được lưu trữ, trang tạo mới công văn, trang quản lý công văn, trang chỉnh sửa các công văn đã lưu trữ.

* 1. **Giao diện trang chủ**

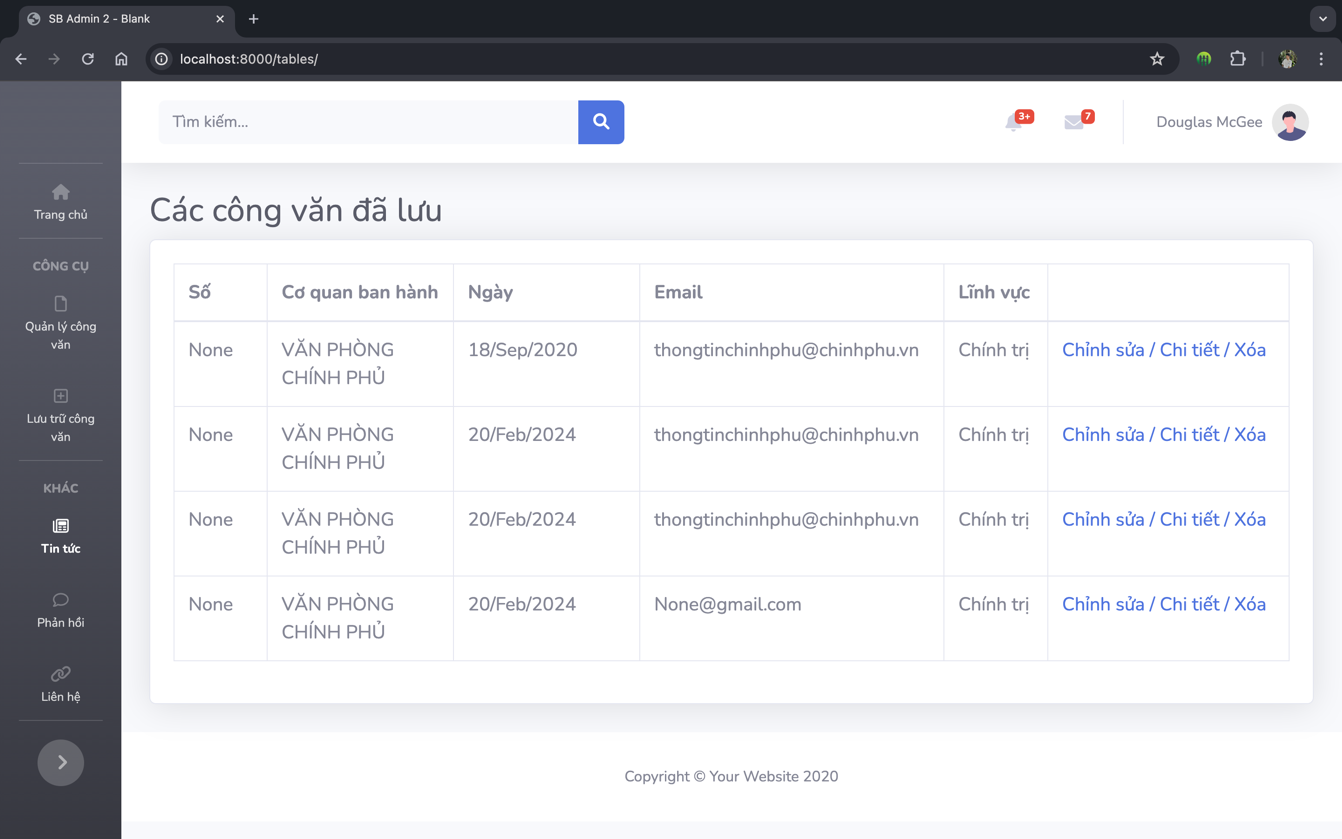
Khi người dùng truy cập vào website thì trang đầu tiên sẽ là trang chủ. ở trang chủ website sẽ hiển thị các thông tin cho người dùng biết, bên cạnh đó người dùng có thể dùng các tính năng mà trang web cung cấp ở thanh slidebar.



Hình 6: Giao diện trang chủ

* 1. **Giao diện trang quản lý công văn**

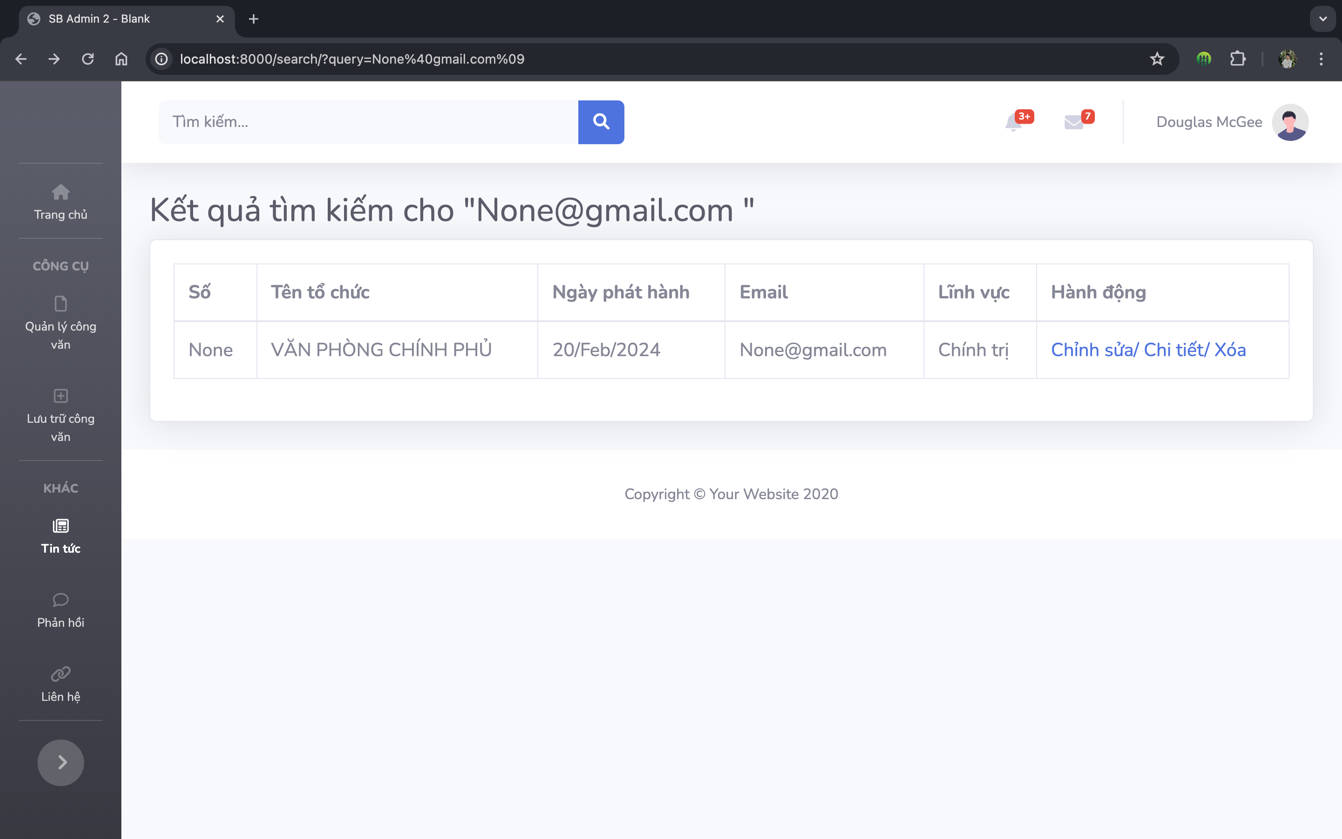
Khi người dùng cần xem những công văn đã lưu trữ thì có thể truy cập vào chức năng quản lý công văn. ở đây sẽ hiển thị các công văn đã được lưu trữ, từ đó có thể tùy chỉnh công văn, xem chi tiết, tải xuống và xóa công văn khi cần thiết.



Hình 7: Giao diện trang quản lý

* 1. **Giao diện trang tìm kiếm công văn**

Khi người dùng muốn tìm kiếm công vă, có thể nhập bất kỳ từ khóa nào liên quan đến các trường lưu trữ của công văn lên ô tìm kiếm và sau đó nhấn nút, hệ thống sẽ trả về những kết quả như người dùng mong muốn. Và ở đây người dùng cũng có thể thực hiện những thao tác tương tự như bên trang quản lý công văn.

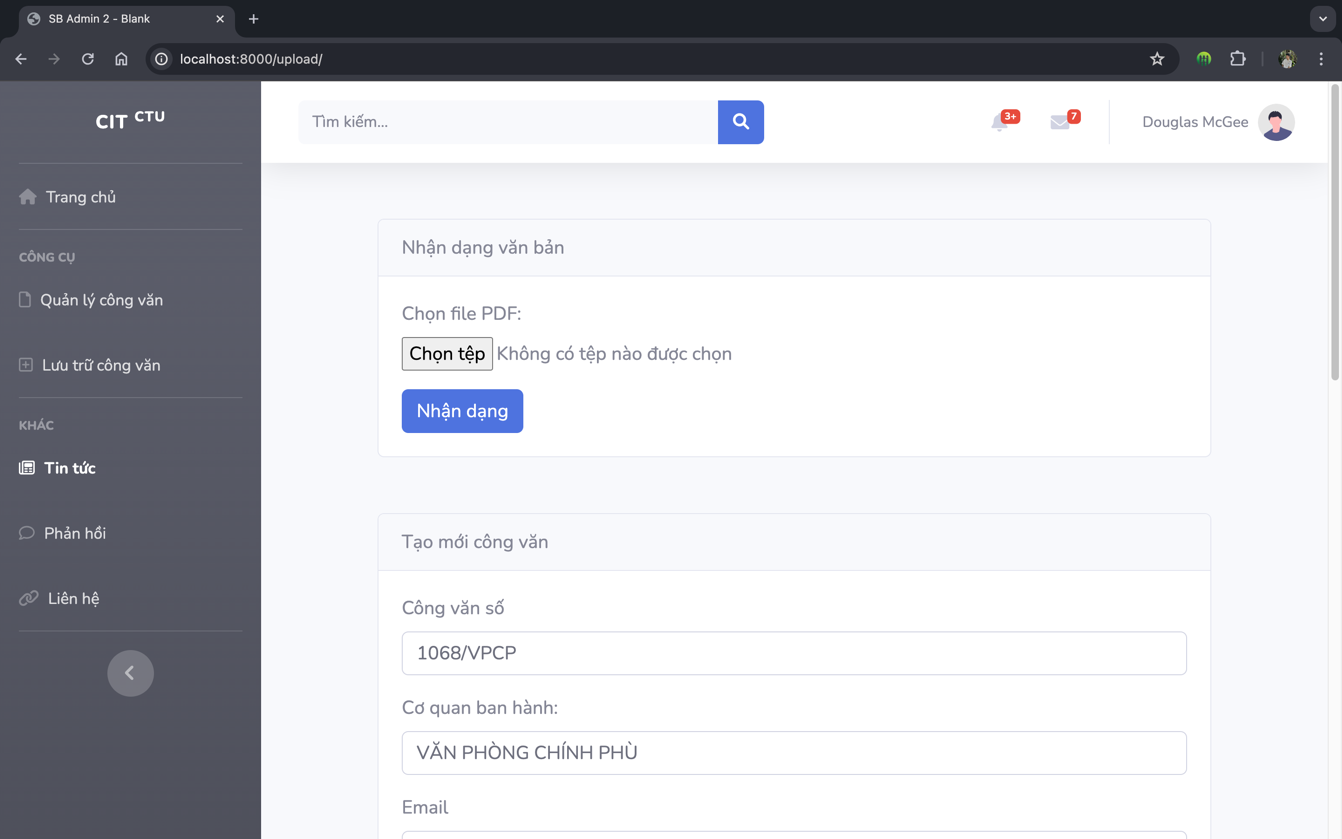


Hình 8: Giao diện trang tìm kiếm

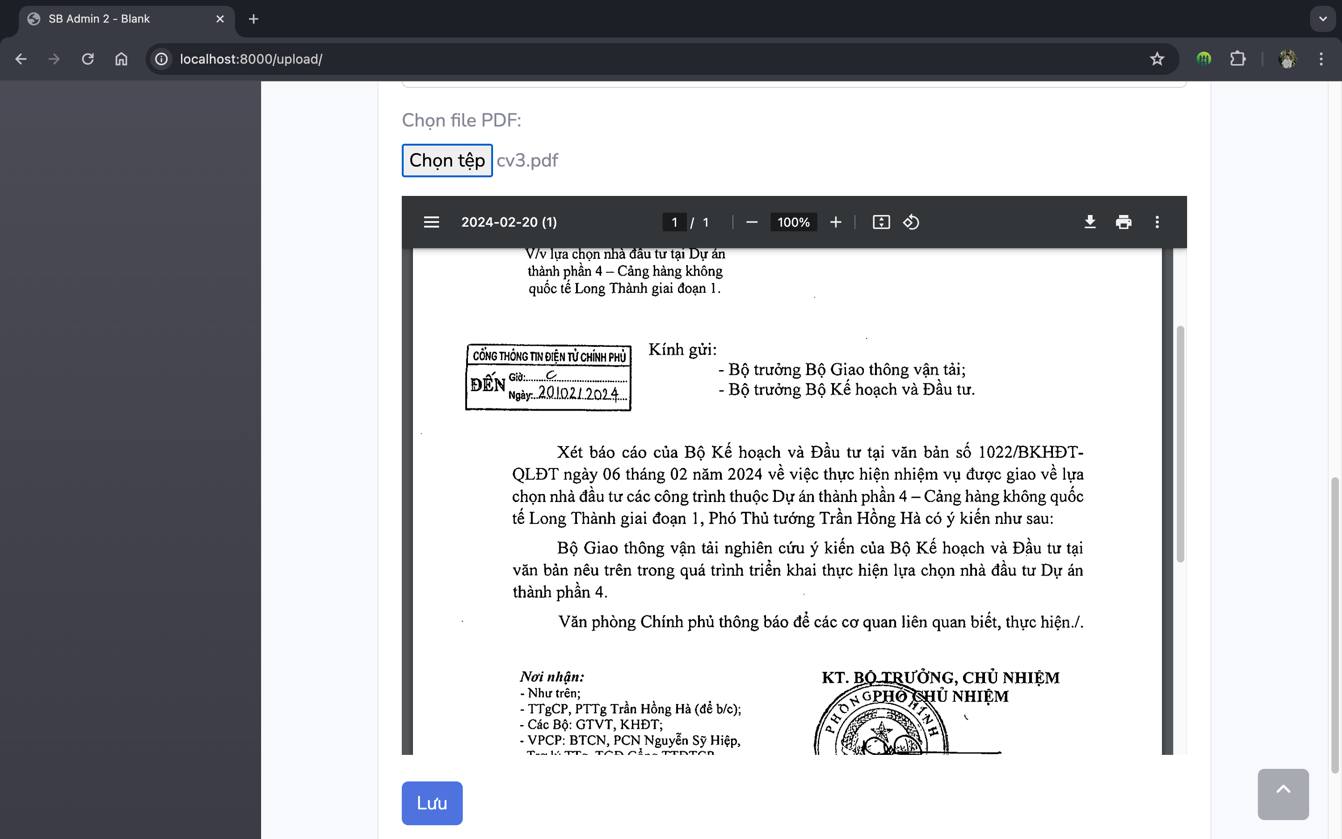
* 1. **Giao diện trang lưu trữ công văn**

Khi người dùng muốn thêm mới công văn để lưu trữ vào hệ thống, người dùng sẽ truy cập vào chức năng thêm mới công văn. ở đây người dùng có thể nhập tay công văn bằng cách điền vào các trường mà form yêu cầu hoặc có thể dùng tính năng nhận dạng văn bản để tự động hóa quá trình điền vào form.

Để sử dụng tính năng nhận dạng người dùng cần tải lên tệp PDF để hệ thống có thể nhận dạng và điền vào form. Nếu người dùng sử dụng tính năng nhận dạng nhưng kết quả không như mong muốn thì có thể sửa lại bằng tay cho đúng. Bên cạnh đó hệ thống còn hỗ trợ lưu trữ file công văn, khi tải công văn lên hệ thống sẽ hiển thị công văn đó để người dùng tiện theo dõi nội dung công văn đã chọn.



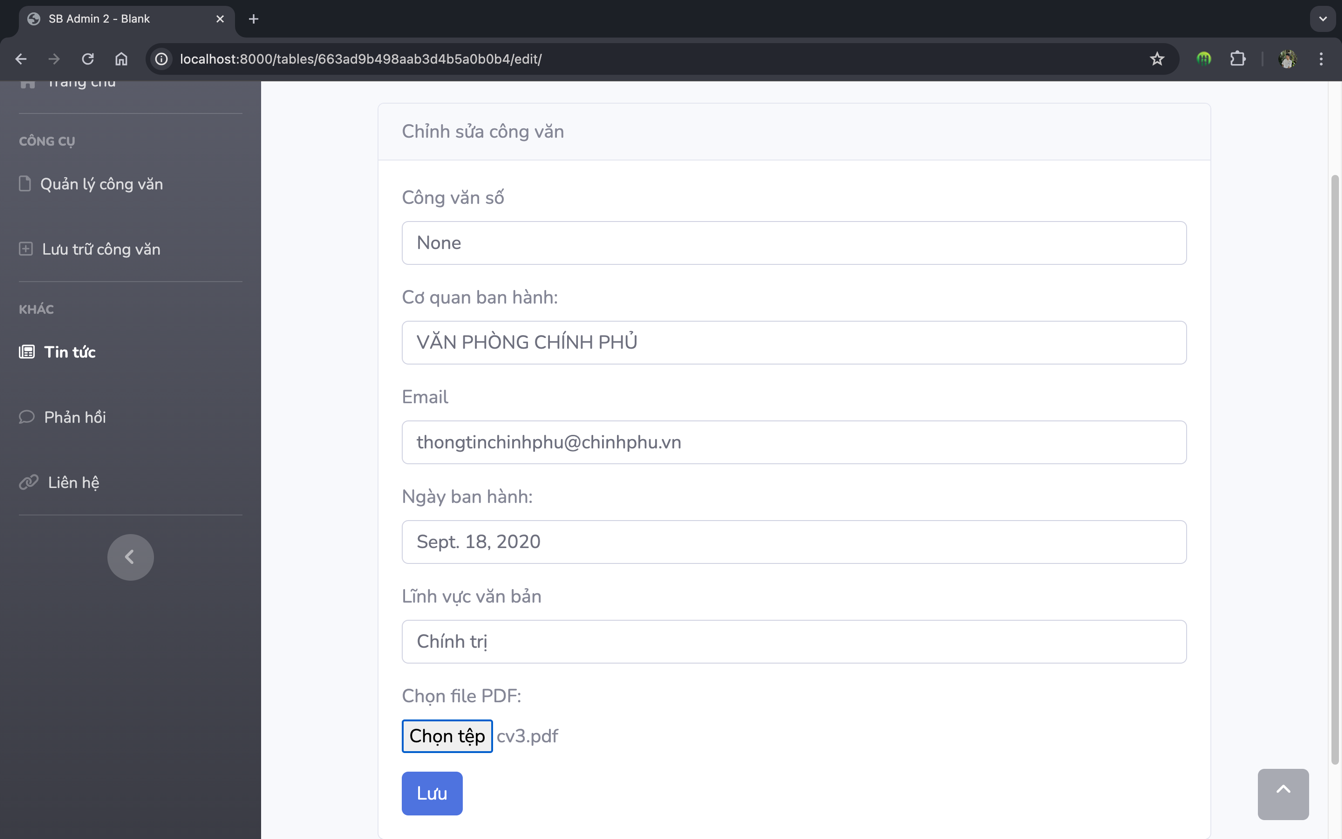
Hình 9: Giao diện trang Upload (nhận dạng)



Hình 10: Giao diện trang Upload (Hiển thị)

* 1. **Giao diện trang chỉnh sửa công văn**

Khi người dùng muốn chỉnh sửa lại các trường thông tin đã lưu trữ, người dùng có thể nhấn vào nút chỉnh sửa để điều chỉnh thông tin đã lưu cho chính xác.



Hình 11: Giao diện trang chỉnh sửa

1. **Kết quả kiểm tra**

Giao diện còn đơn giản, cần bổ sung thêm chức năng đăng nhập, đăng ký để có thể phân quyền người dùng, tránh người dùng thay đổi nội dung công văn tự do gây ảnh hưởng tới chất lượng công việc. trong hệ thống có sẵn 20 công văn được lưu trữ sẵn được tìm từ các nguồn như trang web chinhphu.vn. Số lượng công văn lưu trữ còn khá ít. Chưa có phần hậu xử lý khi đã nhận dạng được văn bản từ Tesseract dẫn đến còn sai lỗi chính tả khá nhiều, độ chính xác không cao. Một vài công văn còn nhận dạng sai nội dung hoặc không rút trích được thông tin cần thiết.

# PHẦN KẾT LUẬN

1. **Kết quả đạt được**

Xây dựng được website quản lý công văn với những chức năng cơ bản như thêm, sửa, xóa, tải về, tìm kiếm công văn. Bên cạnh đó tích hợp thành công hệ thống nhận dạng văn bản để tự động hóa quá trình nhập liệu lưu trữ công văn. Giao diện thiết kế đơn giản dễ sử dụng đảm bảo các chức năng cơ bản của một website quản lý công văn

1. **Hướng phát triển**

Website còn chưa có chức năng đăng nhập, đăng ký, bảo mật thông tin của dữ liệu. cần phải có những giải pháp giúp bảo mật thông tin để tránh bị lộ thông tin ra bên ngoài.

Ngoài ra cần phân quyền các tài khoản để dễ dàng quản lý, tránh thông tin bị sửa đổi mất kiểm soát gây ra sai lệch về thông tin.

Thư viện Tesseract OCR còn nhận dạng sai rất nhiều các ký tự, do đó cần phải có một quá trình xử lý văn bản sau khi nhận dạng để tăng độ chính xác của dữ liệu khi điền vào form. Hạn chế nhiều về mặt font chữ cần thu thập dữ liệu để huấn luyện mô hình nhận dạng được nhiều font chữ hơn.

Giao diện website còn khá đơn giản, chưa có những tính năng nâng cao giúp người dùng tiện lợi hơn trong việc quản lý công văn.

Tối ưu hóa hệ thống tìm kiếm và thêm các bộ lọc để lọc được thông tin một cách chính xác hơn.

Tích hợp chatbox để hỗ trợ trả lời tự động về các thông tin của công văn.

Có thể tích hợp các thư viện tự động hóa sửa lỗi chính tả và rút trích thông tin đặc trưng của văn bản như: Underthesea, vnCoreNLP,…

Nghiên cứu thêm các mô hình khác hỗ trợ tiếng Việt tốt hơn Tesseract như VietOCR để xảy ra ít lỗi hơn trong quá trình huấn luyện.

**TÀI LIỆU KHAM KHẢO**

Các tài liệu kham khảo

[1] Lê Duy Khang (B1913307) “*Xây dựng trang website quản lý công văn có tích hợp chức năng nhận diện ký tự quang học*” (2023)

[2] Priya Sharm, Randhir Singh “Performace of English Character Recognition with and without Noise, India” (2013)

# [3] Shunji mori, Hirobumi Nishida, Hiromitsu Yamada “Optical Character Recognition” (1999)

Các website kham khảo:

[4] www.djangoproject.com

[5] www.miai.vn

[6] chinhphu.vn

[7] www.mongdb.com