

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC CẦN THƠ  
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG



LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC  
NGÀNH CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI  
**XÂY DỰNG HỆ THỐNG LẤY TIN VÀ  
PHÂN LOẠI TIN TỰ ĐỘNG**

Hồ Thanh Sang – MSSV: B1507150  
Nguyễn Thanh Toàn – MSSV: B1507174  
Khóa: K41

Cần Thơ, Tháng 12/2019

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC CẦN THƠ**  
**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**  
**NGÀNH CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI**

**XÂY DỰNG HỆ THỐNG LẤY TIN VÀ**  
**PHÂN LOẠI TIN TỰ ĐỘNG**

**Giáo viên hướng dẫn:**

Ths.GVC. Võ Huỳnh Trâm

**Sinh viên thực hiện:**

Hồ Thanh Sang – B1507150

Nguyễn Thanh Toàn – B1507174

**Cần Thơ, Tháng 12/2019**

## This image shows a full page of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page, providing a template for handwriting practice. There are no margins, text, or other markings on the page.

## This image shows a full page of a handwriting practice worksheet. It consists of numerous horizontal rows, each defined by two parallel dotted lines. The rows are evenly spaced and extend across the entire width of the page, providing a guide for letter height and placement. There is no text or other markings on the page.

## LỜI CẢM ƠN

Chúng em xin bày tỏ lòng biết ơn sâu sắc của mình đến cô Võ Huỳnh Trâm, thuộc bộ môn Kỹ Thuật Phần Mềm, khoa Công Nghệ Thông Tin, trường Đại học Cần Thơ. Trong quá trình thực hiện luận văn, cô đã nhiệt tình giúp đỡ, giải đáp các thắc mắc tạo động lực giúp chúng em hoàn thành luận văn tốt nghiệp này. Chúng em cũng xin được bày tỏ lời cảm ơn tới các thầy cô trong bộ môn nói riêng và trong khoa Công nghệ thông tin nói chung đã nhiệt tình giảng dạy để giúp chúng em có được như ngày hôm nay. Cuối cùng là lời cảm ơn tới gia đình, bạn bè những người luôn sát cánh bên cạnh những lúc khó khăn, luôn ủng hộ giúp đỡ để hoàn thành khóa luận này.

Cần Thơ, ngày 4 tháng 12 năm 2019

Sinh viên thực hiện

Hồ Thanh Sang

Nguyễn Thanh Toàn

## MỤC LỤC

LỜI CẢM ƠN.....	i
MỤC LỤC .....	ii
MỤC LỤC HÌNH ẢNH.....	vi
DANH MỤC BIỂU BẢNG.....	viii
CÁC TỪ VIẾT TẮT VÀ CÁC ĐỊNH NGHĨA .....	x
TÓM TẮT .....	xiii
ABSTRACT .....	xiv
CAM KẾT KẾT QUẢ .....	xv
PHẦN GIỚI THIỆU .....	1
I.    ĐẶT VẤN ĐỀ.....	1
II.   LỊCH SỬ GIẢI QUYẾT VẤN ĐỀ.....	1
III.  MỤC TIÊU ĐỀ TÀI.....	2
IV.  BỐ CỤC LUẬN VĂN .....	3
V.   ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU .....	3
1.  Đối tượng nghiên cứu.....	3
2.  Phạm vi nghiên cứu.....	3
3.  Nội dung nghiên cứu .....	3
4.  Quy trình nghiên cứu.....	4
VI.  CÔNG NGHỆ SỬ DỤNG TRONG ĐỀ TÀI.....	4
1.  Ngôn ngữ lập trình Python .....	4
2.  Docker .....	4
3.  Scrapy và Web crawler.....	4
4.  RSS là gì ? Tại sao dùng RSS để crawl tin tức.....	5
5.  Nodejs – Express framework.....	5
6.  Flask .....	5
7.  MongoDB .....	5
8.  Ghost CMS .....	6

9. Spiderkeeper .....	6
10. SpaCy.....	6
11. Mobiledoc .....	6
PHẦN NỘI DUNG .....	8
CHƯƠNG I MÔ TẢ BÀI TOÁN.....	8
I. MÔ TẢ HỆ THỐNG .....	8
II. MÔ TẢ CHỨC NĂNG HỆ THỐNG.....	9
1. Sơ đồ chức năng.....	9
2. Mô tả chức năng.....	9
III. CÁC YÊU CẦU GIAO TIẾP .....	11
1. Giao tiếp phần cứng .....	11
2. Giao tiếp phần mềm .....	11
3. Giao tiếp truyền thông.....	11
IV. CÁC YÊU CẦU PHI CHỨC NĂNG .....	11
1. Yêu cầu thực thi .....	11
2. Yêu cầu an toàn.....	11
3. Các đặc điểm chất lượng phần mềm .....	12
4. Các yêu cầu khác.....	12
V. LỰA CHỌN VÀ ĐÁNH GIÁ GIẢI PHÁP .....	12
CHƯƠNG II THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP.....	14
I. PHÂN HỆ I : XÂY DỰNG HỆ THỐNG LẤY TIN TỰ ĐỘNG.....	14
1. Tổng quan hệ thống.....	14
2. Kiến trúc hệ thống.....	15
3. Thiết kế dữ liệu .....	20
4. Thiết kế hệ thống lấy tin tự động.....	22
II. PHÂN HỆ II : XÂY DỰNG HỆ THỐNG PHÂN LOẠI TIN TỰ ĐỘNG	31
1. Tổng quan hệ thống.....	31

2. Kiến trúc hệ thống.....	32
3. Thiết kế hệ thống phân loại tin .....	41
III. HỆ THỐNG HIỂN THỊ KẾT QUẢ.....	49
1. Tổng quan hệ thống.....	49
2. Kiến trúc hệ thống.....	50
3. Thiết kế hệ thống của quá trình hiển thị kết quả .....	53
IV. TỰ ĐỘNG HÓA HỆ THỐNG .....	54
CHƯƠNG III KIỂM THỬ VÀ ĐÁNH GIÁ .....	57
I. GIỚI THIỆU .....	57
1. Mục tiêu kiểm thử và đánh giá .....	57
2. Phạm vi kiểm thử .....	57
II. CHI TIẾT KẾ HOẠCH KIỂM THỬ .....	57
1. Các chức năng sẽ được kiểm thử .....	57
2. Các chức năng không được kiểm thử.....	57
3. Cách tiếp cận.....	58
4. Tiêu chí kiểm thử thành công / thất bại.....	58
5. Tiêu chí đình chỉ và yêu cầu bắt đầu làm lại .....	58
6. Sản phẩm bàn giao của kiểm thử .....	58
III. CÁC TRƯỜNG HỢP KIỂM THỬ .....	59
1. Trường hợp kiểm thử 1: Hoạt động hệ thống lấy tin. ....	59
2. Trường hợp kiểm thử 2: Độ chính xác của nội dung tin tức. ....	60
3. Trường hợp kiểm thử 3: Khả năng phân loại của hệ thống.....	61
4. Trường hợp kiểm thử 4: Trang hiển thị kết quả .....	62
5. Trường hợp kiểm thử 5 : Tự động hóa hệ thống .....	63
IV. KẾT QUẢ KIỂM THỬ .....	64
PHẦN KẾT LUẬN.....	65
I. KẾT QUẢ ĐẠT ĐƯỢC .....	65
1. Về lý thuyết.....	65



2. Về chương trình .....	65
3. Khả năng ứng dụng thực tiễn.....	65
II. HẠN CHẾ.....	66
III. HƯỚNG PHÁT TRIỂN .....	66
TÀI LIỆU THAM KHẢO.....	67
PHỤ LỤC .....	69
I. CÁC CÔNG CỤ CHÍNH TRONG HỆ THỐNG.....	69
1. Ngôn ngữ lập trình Python .....	69
2. Docker .....	71
3. Scrapyd và Web Crawler.....	72
4. RSS.....	74
5. Postman .....	75
II. HƯỚNG DẪN CÀI ĐẶT VÀ SỬ DỤNG.....	80
1. Cài đặt Visual Studio Code .....	80
2. Cài đặt Python.....	85
3. Cài đặt Docker .....	86
4. Cài đặt thư viện spaCy .....	88
5. Cài đặt gói nhận dạng ngôn ngữ tiếng Việt cho spaCy.....	89
6. Cài đặt Nodejs.....	89
7. Cài đặt Nodejs express .....	92
8. Cài đặt thư viện Flask.....	93
9. Cài đặt MongoDB trong Docker.....	94
10. Cài đặt Ghost CMS .....	94
11. Cài đặt SpiderKeeper.....	95
12. Cài đặt Scrapy .....	95

## MỤC LỤC HÌNH ẢNH

Hình 1 Sơ đồ chức năng của hệ thống .....	9
Hình 2 Lưu đồ hoạt động của hệ thống.....	10
Hình 3 Kiến trúc Scrapy .....	15
Hình 4 Sơ đồ chức năng hệ thống Nodejs.....	17
Hình 5 Cấu trúc tập tin RSS .....	18
Hình 6 Sơ đồ hoạt động của quá trình lấy tin.....	19
Hình 7 Cấu trúc dữ liệu trong MongoDB .....	21
Hình 8 Sơ đồ hoạt động quá trình trích xuất liên kết URL.....	22
Hình 9 Sơ đồ quá trình kiểm tra liên kết URL .....	24
Hình 10 Sơ đồ hoạt động của hàm checkDuplicate.....	26
Hình 11 Sơ đồ quá trình kiểm tra liên kết URL của Nodejs .....	27
Hình 12 Sơ đồ quá trình trích xuất HTML của Scrapy.....	29
Hình 13 Kiến trúc spaCy .....	32
Hình 14 Cấu trúc tập huấn luyện.....	34
Hình 15 Kiến trúc quá trình đào tạo tập huấn luyện của spaCy.....	35
Hình 16 Quá trình thu thập dữ liệu .....	36
Hình 17 Sơ đồ hoạt động quá trình tạo tập huấn luyện .....	37
Hình 18 Sơ đồ hoạt động của hệ thống phân loại tin.....	39
Hình 19 Quá trình mã hóa dữ liệu sang chuẩn URL .....	41
Hình 20 Quá trình phân loại tin tức của Flask .....	43
Hình 21 Quá trình tạo tập huấn luyện.....	45
Hình 22 Quá trình kiểm tra "mẫu" .....	47
Hình 23 Kiến trúc Ghost CMS .....	50
Hình 24 Cấu trúc chuẩn dữ liệu Mobydoc .....	51
Hình 25 Sơ đồ quá trình tải dữ liệu lên Ghost CMS.....	52
Hình 26 Sơ đồ hoạt động của hệ thống hiển thị kết quả .....	53
Hình 27 Các bước tự động hóa hệ thống .....	54

Hình 28 Tải tập tin egg lên spiderkeeper .....	55
Hình 29 Thiết lập thời gian chạy của hệ thống .....	56

## DANH MỤC BIỂU BẢNG

Bảng 1 Mô tả kiến trúc spaCy .....	16
Bảng 2 Mô tả quá trình lấy tin.....	20
Bảng 3 Mô tả thành phần cấu trúc dữ liệu .....	22
Bảng 4 Mô tả quá trình trích xuất liên kết URL.....	23
Bảng 5 Mô tả quá trình kiểm tra liên kết URL.....	25
Bảng 6 Mô tả quá trình hoạt động hàm checkDuplicate.....	26
Bảng 7 Mô tả quá trình kiểm tra liên kết URL của Nodejs server .....	28
Bảng 8 Mô tả quá trình đọc liên kết URL và trích xuất HTML.....	30
Bảng 9 Các thành phần trong kiến trúc spaCy .....	34
Bảng 10 Mô tả quá trình huấn luyện của spaCy.....	36
Bảng 11 Mô tả quá trình tạo tập huấn luyện .....	38
Bảng 12 Mô tả quá trình phân loại tin .....	40
Bảng 13 Mô tả quá trình mã hóa dữ liệu sang URL.....	42
Bảng 14 Mô tả quá trình phân loại tin của Flask.....	44
Bảng 15 Mô tả quá trình tạo tập “mẫu” .....	46
Bảng 16 Mô tả quá trình kiểm tra tập “mẫu” .....	48
Bảng 17 Mô tả quá trình hiển thị kết quả.....	52
Bảng 18 Mô tả hoạt động hệ thống hiển thị kết quả.....	54
Bảng 19 Kết nhập trường hợp kiểm thử 1.....	59
Bảng 20 Kết xuất trường hợp kiểm thử 1 .....	59
Bảng 21 Kết nhập trường hợp kiểm thử 2.....	60
Bảng 22 Kết xuất trường hợp kiểm thử 2 .....	60
Bảng 23 Kết nhập trường hợp kiểm thử 3.....	61
Bảng 24 Kết xuất trường hợp kiểm thử 3 .....	61
Bảng 25 Kết nhập trường hợp kiểm thử 4.....	62
Bảng 26 Kết xuất trường hợp kiểm thử 4 .....	62

Bảng 27 Kết nhập trường hợp kiểm thử 5.....	63
Bảng 28 Kết xuất trường hợp kiểm thử 5 .....	63
Bảng 29 Kết quả kiểm thử.....	64

## CÁC TỪ VIẾT TẮT VÀ CÁC ĐỊNH NGHĨA

Các từ viết tắt / định nghĩa	Mô tả
URL	Uniform Resource Locator (Định vị tài nguyên thống nhất), được dùng để tham chiếu tới tài nguyên trên Internet.
NoSql	Not-Only SQL
SQL	Structured Query Language
API	Application Programming Interface - giao diện lập trình ứng dụng
Mobiledoc	Là một định dạng bài báo hoặc bài viết đơn giản
Web crawler	Là một phần mềm có khả năng tự động lấy dữ liệu như ảnh, text, ... trên WWW.
Machine Learning	Máy học
Website	Là một trang thông tin với mục đích là để giới thiệu, cập nhật những thông tin về các doanh nghiệp, sản phẩm, hoạt động cũng như tin tức, chia sẻ bí quyết,... để phát triển thương hiệu.
RSS	Really Simple Syndication (hay Rich Site Summary) – Cung cấp thông tin thực sự đơn giản.
Blog	Là một website thông tin riêng hoặc nhật ký trực tuyến, với cách trình bày các bài viết mới nhất được đưa lên đầu.
Framework	Là một bộ khung cung cấp các chức năng, giải pháp được cài đặt sẵn giúp tiết kiệm thời gian trong quá trình phát triển ứng dụng.

Middleware	Là những đoạn mã trung gian nằm giữa các request và response. Nó nhận các request, thi hành các mệnh lệnh tương ứng trên request đó.
Request	Là yêu cầu từ client lên server
Response	Là server trả kết quả về cho client
Server	Máy chủ
Client	Máy khách
CSDL	Cơ sở dữ liệu
CMS	Content Management System – Hệ thống quản trị nội dung
Open source	Mã nguồn mở
Service	Dịch vụ
Spider	Là thành phần thực hiện nhiệm vụ lấy tin trong web crawler
NLP	Neuro Linguistic Programming (Lập Trình Ngôn Ngữ Tư Duy)
Deep – learning	Là một phạm trù nhỏ của machine learning, deep learning tập trung giải quyết các vấn đề liên quan đến trí thông minh nhân tạo nhằm nâng cấp các công nghệ như nhận diện giọng nói, tầm nhìn máy tính và xử lý ngôn ngữ tự nhiên.
AI	Artificial Intelligence – Trí thông minh nhân tạo
HTML	HyperText Markup Language - Ngôn ngữ Đánh dấu Siêu văn bản
Kernel	Nhân của hệ điều hành

REST	Hiểu đơn giản nó là một bộ các ràng buộc và quy ước , khi áp dụng đầy đủ vào hệ thống của bạn thì ta có 1 hệ thống REST
ORM	Object Relational Mapping” đây là tên gọi chỉ việc ánh xạ các record dữ liệu trong hệ quản trị cơ sở dữ liệu sang dạng đối tượng mà mã nguồn đang định nghĩa trong class.
Admin DashBoard	Khu vực dùng để truy cập và quản trị website.



## TÓM TẮT

Do nhu cầu thu thập thông tin của con người ngày càng tăng, lượng thông tin trên Internet ngày càng phức tạp nên vấn đề tổng hợp thông tin ngày càng trở nên bức thiết. Với một lượng dữ liệu lớn việc thu thập bằng tay sẽ tốn rất nhiều công sức, và không đạt hiệu quả cao, chính vì thế một công cụ hỗ trợ tổng hợp thông tin tự động là trình thu thập web (web crawler) đã ra đời. Đề tài luận văn đặt ra vấn đề tìm hiểu về trình thu thập web và bước đầu sẽ xây dựng một ứng dụng có khả năng tổng hợp và phân loại tin tức tự động từ nhiều trang báo điện tử khác nhau. Ứng dụng được viết bằng ngôn ngữ lập trình Python và được xây dựng dựa trên các tiêu chí: tốc độ thu thập nhanh, hệ thống hoạt động tự động, cơ sở dữ liệu gọn nhẹ, đảm bảo tính toàn vẹn của tài liệu gốc.

Hệ thống thu thập và phân loại tin tự động gồm hai quá trình chính là thu thập và phân loại tin tức. Nội dung luận văn được chia thành hai phân hệ tương ứng với hai quá trình trên là: Phân hệ xây dựng hệ thống lấy tin tự động và phân hệ xây dựng hệ thống phân loại tin tự động. Các chức năng chính của hệ thống là lấy tin, phân loại tin, đưa tin tức lên trang hiển thị. Điểm nổi bật của hệ thống là vận hành tự động, hệ thống sẽ luôn cập nhật tin tức mới nhất của những trang tin tức nguồn vì chúng ta có thể thiết đặt được thời gian chạy của hệ thống. Nội dung tin tức lấy được có độ chính xác cao đối với những trang tin gồm chữ và hình ảnh. Ngoài ra hệ thống có khả năng phân biệt và loại bỏ những bài tin bị trùng lặp (với những tin trong cùng trang nguồn). Hệ thống phân loại tin sẽ luôn cập nhật lại tập huấn luyện (train) mỗi khi có một bài báo mới được duyệt qua hệ thống, vì thế hệ thống phân loại tin tự động sẽ ngày càng được thông minh và chính xác hơn. Các công cụ hỗ trợ được thiết kế đơn giản và phù hợp với hệ thống nên cho tốc độ hoạt động rất nhanh, cơ sở dữ liệu của hệ thống gọn nhẹ.

Hệ thống đã đạt được mục tiêu đề ra ban đầu là hoạt động ổn định, tốc độ xử lý nhanh, nội dung bài viết lấy về khá đầy đủ, có một vài trường hợp bài tin đặc biệt thì không xử lý được. Hệ thống phân loại bằng máy học cũng đáp ứng được yêu cầu đề ra là có khả năng phân loại tin theo tiêu đề, độ chính xác của kết quả phân loại đạt được mục tiêu mong đợi. Về hạn chế thì hệ thống sử dụng Ghost CMS để hiển thị kết quả nên về kiểu chữ cũng như các tùy chỉnh trang hiển thị không được bắt mắt, giao diện hiển thị còn đơn giản. Trong tương lai, dự kiến hệ thống sẽ mở rộng thêm nguồn lấy tin và phát triển thêm chức năng chống trùng lặp giữa các nguồn tin khác nhau.

## ABSTRACT

Due to the increasing demand for information gathering of people and the increasing amount of information on the Internet, the issue of information synthesis has become increasingly urgent. With a large amount of data collected manually, it is very laborious, and not very effective, so a tool to support automatic information synthesis is a web crawler was born. The thesis topic discuss about web crawler and will initially build an application that automatically synthesizes and categorizes news from various websites. The application is written in Python programming language and is built on the criteria: fast collection speed, automatic operation system, lightweight database, ensuring the integrity of the original document.

The automatic information collection and classification system consists of two main processes: news collection and classification. The thesis content is divided into two modules corresponding to the above two processes: Construction module of automatic information collection system and construction of automatic information classification system. The main functions of the system is taking news, categorizing news, putting news on display page. The highlight of the system is automatic operation, the system will always update the latest news of source news sites because we can set the system's runtime. News content obtained with high accuracy for news sites including text and images. In addition, the system has the ability to distinguish and remove duplicate news (with news in the same source page). The classification system will always update the training model whenever a new article is approved by the system, so the automatic information classification system will more and more intelligent and accurate. The support tools are designed to be simple and suitable for the system, so the operation speed is very fast, the database of the system is compact.

The system has achieved the initial goal of stable operation, fast processing speed, the content of the article is quite sufficient, there are some special articles that cannot be processed. The classification system also meets the requirements of being able to classify information by title, the accuracy of classification results to achieve the expected goal. In terms of limitations, the system uses Ghost CMS to display the results, so the font style as well as the page display settings are not eye-catching, the display interface is still simple. In the future, it is expected that the system will expand its sources of information and develop the function of preventing duplication between different sources.

## CAM KẾT KẾT QUẢ

Chúng tôi xin cam kết luận văn này được hoàn thành dựa trên kết quả nghiên cứu của nhóm và các kết quả này của nghiên cứu chưa được dùng cho bất cứ luận văn cùng cấp nào khác.

Chúng tôi xin cam đoan mọi sự giúp đỡ cho việc thực hiện luận văn này đã được cảm ơn và các tài liệu tham khảo trong luận văn đã được ghi rõ nguồn gốc.

Cần Thơ, ngày 4 tháng 12 năm 2019

Sinh viên thực hiện

Hồ Thanh Sang

Nguyễn Thanh Toàn

## PHẦN GIỚI THIỆU

### I. ĐẶT VẤN ĐỀ

Ngày nay nhờ sự bùng nổ của công nghệ thông tin, lịch sử nhân loại đã bước sang một trang mới. Những thành tựu của ngành công nghệ thông tin là vô cùng to lớn, nó đã chi phối và làm thay đổi mọi mặt của đời sống xã hội, làm cho cuộc sống của con người văn minh, hiện đại hơn. Sự ra đời của Internet chính là bước tiến vĩ đại của nhân loại, là yếu tố quan trọng bậc nhất chi phối cuộc sống của chúng ta ngày nay. Nhờ có Internet thế giới trở nên ‘phẳng’ hơn, ở mọi nơi trên trái đất chúng ta đều có thể học tập và tìm kiếm thông tin. Theo vòng quay của cuộc sống, thế giới Internet ngày càng rộng lớn và phong phú hơn. Cứ mỗi ngày trôi qua có thêm hàng nghìn trang web được sinh ra để làm giàu cho vốn tài nguyên tri thức của nhân loại. Nhưng cũng chính vì thế mà việc chọn lọc, tìm kiếm thông tin lại trở nên khó khăn hơn. Với kho dữ liệu đồ sộ như Internet, vấn đề trích xuất và tổng hợp thông tin đã trở thành vấn đề thực sự cấp thiết hiện nay. Nếu giải quyết được vấn đề này chúng ta sẽ loại bỏ được một chướng ngại lớn trên con đường tổng hợp thông tin của nhân loại. Đề tài khóa luận đặt ra vấn đề tìm hiểu về trình thu thập thông tin trên web và bước đầu sẽ xây dựng một ứng dụng có khả năng tổng hợp thông tin tự động từ các trang báo điện tử lớn. Đề tài nếu thành công sẽ là bước đi không nhỏ giúp cho việc tổng hợp thông tin trở nên đơn giản hơn, giảm được nhiều chi phí công sức so với việc tổng hợp thủ công.

### II. LỊCH SỬ GIẢI QUYẾT VẤN ĐỀ

**Trong khoa:** Trước đây chưa có luận văn hay chuyên đề nào đề cập đến vấn đề lấy tin từ Internet và phân loại tin tự động bằng máy học.

**Trong nước:** Có một số trang tin tức sử dụng hệ thống lấy tin tự động:

- Báo mới (<https://baomoi.com/>)
- Thanh niên (<https://thanhnien.vn/>)

### III. MỤC TIÊU ĐỀ TÀI

Nội dung của luận văn sẽ tập trung vào các mục tiêu chính sau:

- Đưa ra được một cái nhìn tổng quát về trình thu thập web (web crawler). Xây dựng một web crawler cơ bản.
- Xây dựng một hệ thống phân loại tin tức tự động dựa vào nguồn tin từ web crawler trên.

Để giải quyết các mục tiêu đề trên, nội dung chính luận văn được chia thành các chương lớn:

- Chương 1: Mô tả bài toán
  - Mô tả chi tiết bài toán: Giúp người đọc hiểu rõ các chức năng/tính năng/đặc điểm của sản phẩm/phần mềm/hệ thống/giải pháp là mục đích cần đạt được của đề tài. Mô tả những vấn đề mà giải pháp sẽ xử lý, cải tiến, khắc phục ...
  - Phân tích đánh giá các giải pháp/ có liên quan đến bài toán. Tiếp cận giải quyết vấn đề, chọn lựa giải pháp
- Chương 2: Thiết kế và cài đặt giải pháp : Chương này sẽ chia thành 2 phân hệ lớn:
  - Phân hệ I : Xây dựng hệ thống lấy tin tự động (web crawler) ( Nguyễn Thanh Toàn phụ trách).
  - Phân hệ II: Xây dựng hệ thống phân loại tin tức tự động ( Hồ Thanh Sang phụ trách).

Mỗi phân hệ gồm các nội dung chính như sau :

- Thiết kế kiến trúc tổng thể của hệ thống, giải thích chức năng của từng thành phần trong hệ thống, các giải thuật xử lý của hệ thống hoặc của một thành phần hệ thống, thiết kế cơ sở dữ liệu,...
- Mô tả cách thức cài đặt thiết kế bằng một ngôn ngữ lập trình cụ thể/hệ điều hành/ phần cứng ....
- Chương 3: Kiểm thử và đánh giá: Mô tả mục tiêu kiểm thử, kịch bản kiểm thử và kết quả kiểm thử: có chạy được hay không, chạy đúng không, đạt các mục tiêu đề ra hay không ?

#### **IV. BỐ CỤC LUẬN VĂN**

Nội dung quyền luận văn gồm các phần : phần giới thiệu, phần nội dung, phần kết luận, tài liệu tham khảo và phụ lục.

Phần giới thiệu nêu lên vấn đề cần giải quyết và phạm vi của vấn đề. Qua đó lên kế hoạch và phương pháp thực hiện. Phần giới thiệu bao gồm những nội dung chính: Đặt vấn đề, lịch sử giải quyết vấn đề, mục tiêu đề tài, bố cục của quyền luận văn, đối tượng, phạm vi và nội dung nghiên cứu.

Phần nội dung gồm ba chương, trình bày các nội dung chính của luận văn một cách chi tiết, cách giải quyết và kết quả đạt được. Mô tả bài toán nêu lên chi tiết về bài toán, các chức năng, yêu cầu bài toán đặt ra. Thiết kế và cài đặt giải pháp là mô tả tổng quan hệ thống, thiết kế kiến trúc tổng thể, thiết kế cơ sở dữ liệu, thiết kế giao diện chức năng hệ thống. Kiểm thử và đánh giá gồm có mô tả mục tiêu, kế hoạch, các trường hợp kiểm thử và kết quả kiểm thử, từ đó đưa ra đánh giá đối với các chức năng của hệ thống.

Phần kết luận trình bày kết quả đạt được của đề tài cũng như những hạn chế mà đề tài chưa thực hiện được, ngoài ra đưa ra hướng phát triển sau này.

Phần phụ lục trình bày hướng dẫn cài đặt các công nghệ, thư viện hỗ trợ được sử dụng trong hệ thống.

#### **V. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU**

##### **1. Đối tượng nghiên cứu**

Đối tượng nghiên cứu là những người thường xuyên đọc tin tức trên internet. Những người này có thể là giáo viên, giảng viên, công nhân, người già hoặc người trưởng thành, cũng có thể là trẻ em,...

##### **2. Phạm vi nghiên cứu**

Đề tài nghiên cứu chủ yếu trên internet, nơi có nguồn tài liệu và tin tức vô cùng phong phú và phức tạp.

##### **3. Nội dung nghiên cứu**

Đề tài nghiên cứu được chia thành 2 phần lớn:

- Lấy tin tức từ trên các website tin tức về loại bỏ tin tức trùng lặp (Hồ Thanh Sang phụ trách)
- Phân loại tin tức lấy về và upload lên trang website hiển thị (Nguyễn Thanh Toàn phụ trách)

#### 4. Quy trình nghiên cứu

- Tìm hiểu và thu thập yêu cầu: Thu thập các tài liệu liên quan đến kiến thức cần thiết, tìm hiểu các hệ thống ứng dụng đã có.
- Nghiên cứu và lựa chọn công nghệ sử dụng.
- Đặc tả và thiết kế hệ thống.
- Thiết kế các chức năng chi tiết.
- Lập trình.
- Cài đặt và kiểm thử hệ thống.
- Tổng hợp tài liệu và viết báo cáo.

## VI. CÔNG NGHỆ SỬ DỤNG TRONG ĐỀ TÀI

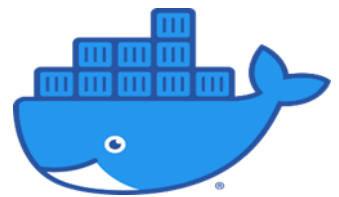
### 1. Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991.



### 2. Docker

Việc setup và deploy application lên một hoặc nhiều server rất vất vả từ việc phải cài đặt các công cụ, môi trường cần cho ứng dụng đến việc chạy được ứng dụng chưa kể việc không đồng nhất giữa các môi trường trên nhiều server khác nhau. Chính vì lý do đó Docker được ra đời để giải quyết vấn đề này.



### 3. Scrapy và Web crawler

Scrapy là một framework được viết bằng Python, nó cấp sẵn 1 cấu trúc tương đối hoàn chỉnh để thực hiện việc lấy và phân tích dữ liệu từ website một cách nhanh chóng và dễ dàng. Scrapy là một công cụ mạnh mẽ để tạo web crawler.



#### 4. RSS là gì ? Tại sao dùng RSS để crawl tin tức

RSS là viết tắt của Really Simple Syndication (hay Rich Site Summary.) – Cung cấp thông tin thực sự đơn giản. RSS cho phép bạn đăng ký ở một blog hoặc website, và khi có thông tin mới từ website bạn sẽ được thông báo mà không phải mất thời gian tìm kiếm.



#### 5. Nodejs – Express framework

Express là một framework giành cho nodejs. Nó cung cấp cho chúng ta rất nhiều tính năng mạnh mẽ trên nền tảng web cũng như trên các ứng dụng di động. Express hỗ trợ các phương thức HTTP và middleware tạo ra một API vô cùng mạnh mẽ và dễ sử dụng. Có thể tổng hợp một số chức năng chính của express như sau:



- Thiết lập các lớp trung gian để trả về các HTTP request.
- Định nghĩa router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
- Cho phép trả về các trang HTML dựa vào các tham số.

#### 6. Flask

Flask là một web frameworks, nó thuộc loại micro-framework được xây dựng bằng ngôn ngữ lập trình Python. Flask cho phép bạn xây dựng các ứng dụng web từ đơn giản tới phức tạp. Nó có thể xây dựng các api nhỏ, ứng dụng web chẳng hạn như các trang web, blog, trang wiki hoặc một website dựa theo thời gian hay thậm chí là một trang web thương mại. Flask cung cấp cho bạn công cụ, các thư viện và các công nghệ hỗ trợ bạn làm những công việc trên.



Flask là một micro-framework. Điều này có nghĩa Flask là một môi trường độc lập, ít sử dụng các thư viện khác bên ngoài. Do vậy, Flask có ưu điểm là nhẹ, có rất ít lỗi do ít bị phụ thuộc cũng như dễ dàng phát hiện và xử lý các lỗi bảo mật.

#### 7. MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.





## 8. Ghost CMS

Giống như WordPress CMS cho PHP, Ghost là một CMS (Content Management System – Hệ thống quản trị nội dung) được viết bằng ngôn ngữ lập trình Nodejs (Javascript). Ghost được sử dụng nhiều để tạo ra các blog, nhưng chúng ta có thể sử dụng, lập trình nó thành các website, shop, ...



Ưu điểm khi sử dụng Ghost CMS làm Website:

- Cài đặt dễ dàng, nhanh chóng
- Quản lý nội dung
- Hỗ trợ lịch đăng bài viết
- Free, Open Source
- Hỗ trợ API để tạo và quản lý bài viết

## 9. Spiderkeeper

Spiderkeeper là một Admin DashBoard được deploy trên Docker, chức năng chính của spiderkeeper là quản lý các file spider (crawler) một cách tự động và ổn định.



Các chức năng của Spiderkeeper:

- Quản lý các spider từ bảng điều khiển. Lên lịch trình để chạy tự động.
- Hiện thị số liệu thống kê sau mỗi lần chạy các spider.
- Cung cấp API quản lý spider.

## 10. SpaCy

SpaCy là một thư viện open-source miễn phí, dùng cho các dự án NLP sử dụng ngôn ngữ python.



SpaCy được thiết kế để giúp bạn xây dựng các ứng dụng xử lý và hiểu được khối lượng lớn văn bản. Nó có thể được sử dụng để xây dựng hệ thống khai thác thông tin hoặc hệ thống hiểu ngôn ngữ tự nhiên hoặc để xử lý trước văn bản dành cho Deep-learning.

## 11. Mobiledoc

Mobiledoc là một định dạng bài báo hoặc bài viết đơn giản nhằm mục đích:

- Nền tảng bất khả tri (Platform agnostic): Có thể kết xuất mà không cần trình phân tích cú pháp HTML.
- Hiệu quả để chuyển đổi (Efficient to transfer): Nén tốt, và hạn chế sự trùng lặp nội dung.
- Mở rộng khi chạy (Extensible at runtime): Lưu trữ nội dung, không bố trí hoặc hiển thị cuối cùng.

## PHẦN NỘI DUNG

### CHƯƠNG I MÔ TẢ BÀI TOÁN

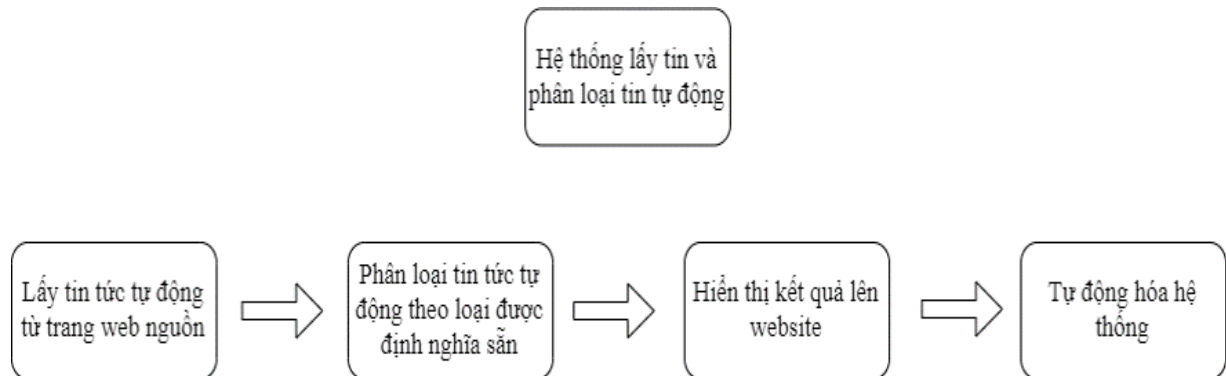
#### I. MÔ TẢ HỆ THỐNG

**Hệ thống lấy tin và phân loại tin tự động** là một hệ thống vận hành tự động có nhiệm vụ thu thập tin tức từ nhiều trang tin tức nguồn, phân loại lại và hiển thị lên trang kết quả. Hệ thống này giúp tổng hợp tin tức một cách nhanh chóng và chính xác, giúp những người thường xuyên đọc báo trực tuyến có thể cập nhật nhanh chóng những tin tức, sự kiện mới nhất vừa xảy ra. Hệ thống còn có chức năng loại bỏ những tin trùng lặp giúp đọc giả tiết kiệm thời gian khi lướt nhiều trang web tin tức luôn gặp phải tin bị trùng lặp. Mỗi trang web tin tức có những ngôn từ để xác định loại tin khác nhau, có thể một bài báo ở trang này nằm ở loại báo “Công Nghệ”, cũng có thể bài đó ở trang khác lại là “Khoa Học” hoặc “Điện Tử”, về mặt ngữ nghĩa thì các từ ngữ trên khác nhau nhưng xét về một dung thì chúng hầu như là một. Hệ thống sử dụng AI, Machine Learning để phân loại lại tin tức giúp định nghĩa lại bài tin thuộc loại cụ thể giúp đọc giả không bị nhầm lẫn về loại bài báo.

Hệ thống sử dụng công nghệ Ghost CMS để làm trang website hiển thị kết quả tin đã phân loại, ưu điểm của Ghost là giao diện thân thiện, dễ sử dụng, đọc giả có thể dễ dàng tìm một bài báo dựa theo các đề mục có sẵn tác giả, loại tin,... Các chức năng của Ghost hầu như đều miễn phí nên đọc giả không cần một tài khoản hay một liên kết đến tài khoản nào khác như Google, Facebook để truy cập vào bài tin. Chỉ cần nhấp vào liên kết bài tin là đọc giả có thể truy cập đầy đủ thông tin cũng như nội dung về bài tin đó. Trang quản trị của Ghost cũng có đầy đủ chức năng như các trang quản trị của các website tin tức lớn, người quản trị có thể quản lý tất cả các bài viết ở trang quản trị, thêm sửa hay xóa bài biết đều thực hiện dễ dàng, Ngoài ra còn có thể chỉnh sửa nội dung của trang chủ Ghost như chèn thêm background, định dạng lại cách hiển thị tin tức (dạng lướt, dạng danh sách,...).

## II. MÔ TẢ CHỨC NĂNG HỆ THỐNG

### 1. Sơ đồ chức năng



Hình 1 Sơ đồ chức năng của hệ thống

### 2. Mô tả chức năng

#### 2.1 Lấy tin tức tự động từ trang web nguồn

Khi hệ thống vận hành, dựa vào thời gian được thiết đặt sẵn bởi spiderkeeper, Hệ thống sẽ tiến hành dò theo đường dẫn RSS để trích xuất ra các đường dẫn HTML, từ các đường dẫn này hệ thống sẽ lấy được nội dung HTML của bài tin và lưu tạm vào bộ nhớ.

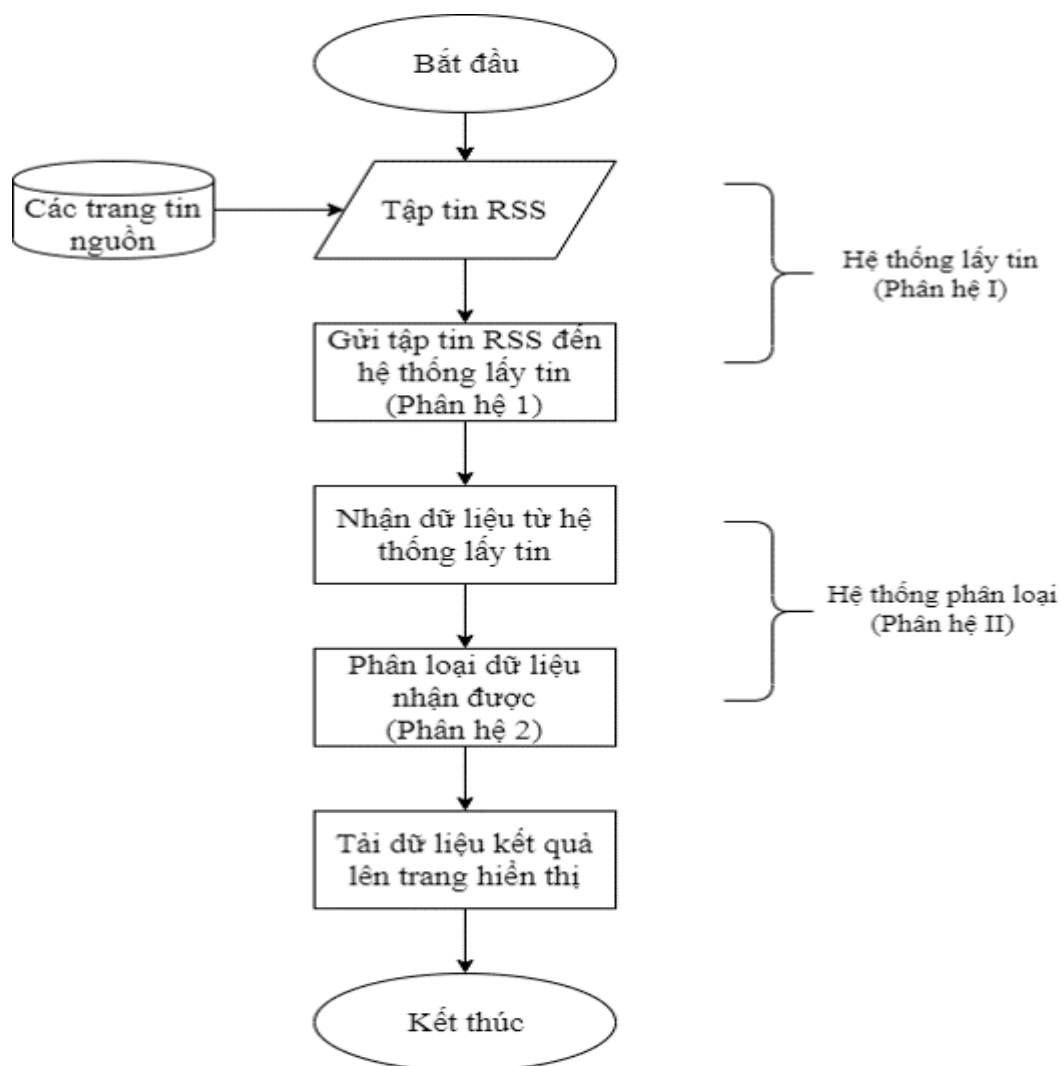
#### 2.2 Phân loại tin tức tự động theo loại được định nghĩa sẵn

Dựa vào nội dung đã lấy được ở trên, hệ thống sẽ tiến hành phân loại nội dung bài tin đó thuộc loại nào, dựa theo tiêu đề và mô tả của bài tin, bài tin sẽ được phân loại tự động theo các loại đã được định nghĩa sẵn trong hệ thống. Nội dung của bài tin sẽ được chuyển từ chuẩn HTML sang chuẩn Mobiledoc để phù hợp với cấu trúc đăng tin của Ghost CMS.

#### 2.3 Hiển thị kết quả lên website

Sau khi bài tin được phân loại và chuyển đổi chuẩn đúng, hệ thống sẽ tiến hành upload lên website hiển thị thông qua các API được Ghost CMS cung cấp.

#### 2.4 Sơ đồ tổng quan của hệ thống



Hình 2 Lưu đồ hoạt động của hệ thống

### **III. CÁC YÊU CẦU GIAO TIẾP**

#### **1. Giao tiếp phần cứng**

Yêu cầu hỗ trợ phần cứng được sử dụng trong hệ thống bao gồm:

Máy tính cá nhân: dùng để sử dụng cho admin, có đầy đủ phần mềm hỗ trợ chạy trang web tin tức và trang quản lý.

Cơ sở dữ liệu: Có thể chứa lượng lớn dữ liệu của trang web, cho phép kết nối nhanh chóng, có thể mở rộng và cải tiến cơ sở dữ liệu trong tương lai khi hệ thống có nhu cầu phát triển thêm chức năng, trang web có thể tương tác với người dùng bằng chuột, màn hình, bàn phím.

#### **2. Giao tiếp phần mềm**

Hệ thống vận hành tốt trên các trình duyệt : Chrome, Firefox,... dữ liệu được quản lý bằng MongoDB và SQLite

Trang web hiển thị chạy ổn định trên máy tính sử dụng các hệ điều hành Windows phiên bản 7, 8, 10, Mac OS, Linux.

#### **3. Giao tiếp truyền thông**

Trang web sử dụng giao thức truyền HTTP, sử dụng API với các giao thức GET, POST để truyền tải dữ liệu giữa hệ thống lấy tin, phân loại tin và website hiển thị.

### **IV. CÁC YÊU CẦU PHI CHỨC NĂNG**

#### **1. Yêu cầu thực thi**

- Thời gian tải dữ liệu của website không quá lâu.
- Hệ thống hoạt động ổn định.
- Hiển thị đầy đủ thông tin nội dung của trang web nguồn.
- Thời gian chạy của hệ thống phải được lên lịch biểu rõ ràng, ổn định.

#### **2. Yêu cầu an toàn**

- Đảm bảo an toàn CSDL trước các nguy hại trên Internet, các hành vi tấn công phá hoại của hacker.
- Sao lưu dữ liệu định kỳ để đảm bảo dữ liệu không bị mất mát.

### 3. Các đặc điểm chất lượng phần mềm

- Có thể chạy tốt trên các trình duyệt phổ biến hiện nay như Chrome, FireFox, Safari, Edge, Cốc Cốc, ...
- Giao diện thân thiện với người dùng: màu sắc hài hoà, kiểu chữ rõ ràng dễ đọc, bố cục thông tin hợp lý.
- Phải hiển thị được trên các thiết bị cầm tay như di động, máy tính bảng.
- Các nút lệnh sắp xếp hợp lý.
- Tốc độ xử lý tốt, chính xác.
- Mức độ bảo mật cao, tin cậy.

### 4. Các yêu cầu khác

- Yêu cầu thiết kế hệ thống sử dụng ngôn ngữ có khả năng đa nền tảng như các ngôn ngữ thiết kế web.
- CSDL tạo ra phải rõ ràng, mạch lạc, tránh dư thừa dữ liệu. Dữ liệu dễ dàng quản lý và truy xuất, đảm bảo an toàn và bảo mật thông tin.
- Ngôn ngữ chính là tiếng Việt trừ một số thuật ngữ không thể thay thế.
- Câu chữ phải đúng chính tả, phù hợp văn hóa, phong tục, tập quán Việt Nam, sử dụng ngôn ngữ phổ thông, không sử dụng tiếng lóng, tiếng địa phương.
- Hệ thống website phải đảm bảo nội dung trong sáng, lành mạnh, hợp pháp, không chứa các tin sai lệch, không liên quan đến các vấn đề nhạy cảm như chính trị, phản động, nội dung phải phù hợp với pháp luật và các quy định.
- Dễ bảo trì cũng như nâng cấp.

## V. LỰA CHỌN VÀ ĐÁNH GIÁ GIẢI PHÁP

Giải pháp để xây dựng trang web bao gồm các bước sau:

- Lựa chọn ngôn ngữ xây dựng hệ thống: Hệ thống sử dụng ngôn ngữ lập trình chính là Python, ngoài ra còn sử dụng Nodejs để xây dựng công cụ hỗ trợ cho hệ thống.
- Sử dụng Ghost CMS làm website hiển thị kết quả cho hệ thống,
- Sử dụng MongoDB và SQLite để quản lý cơ sở dữ liệu cho hệ thống

Đánh giá giải pháp:

- Hệ thống hoạt động ổn định và mượt mà
- Ngôn ngữ sử dụng chủ yếu là Python nên việc quản lý code cũng như chỉnh sửa hệ thống dễ dàng cho lập trình viên.
- Sử dụng Ghost CMS làm website hiển thị tiết kiệm thời gian và công sức hơn so với xây dựng trang web từ php thuần.
- CSDL MongoDB và SQLite miễn phí và có thể chứa nguồn dữ liệu lớn phù hợp cho mở rộng CSDL trong tương lai.



## CHƯƠNG II THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP

### I. PHÂN HỆ I : XÂY DỰNG HỆ THỐNG LẤY TIN TỰ ĐỘNG

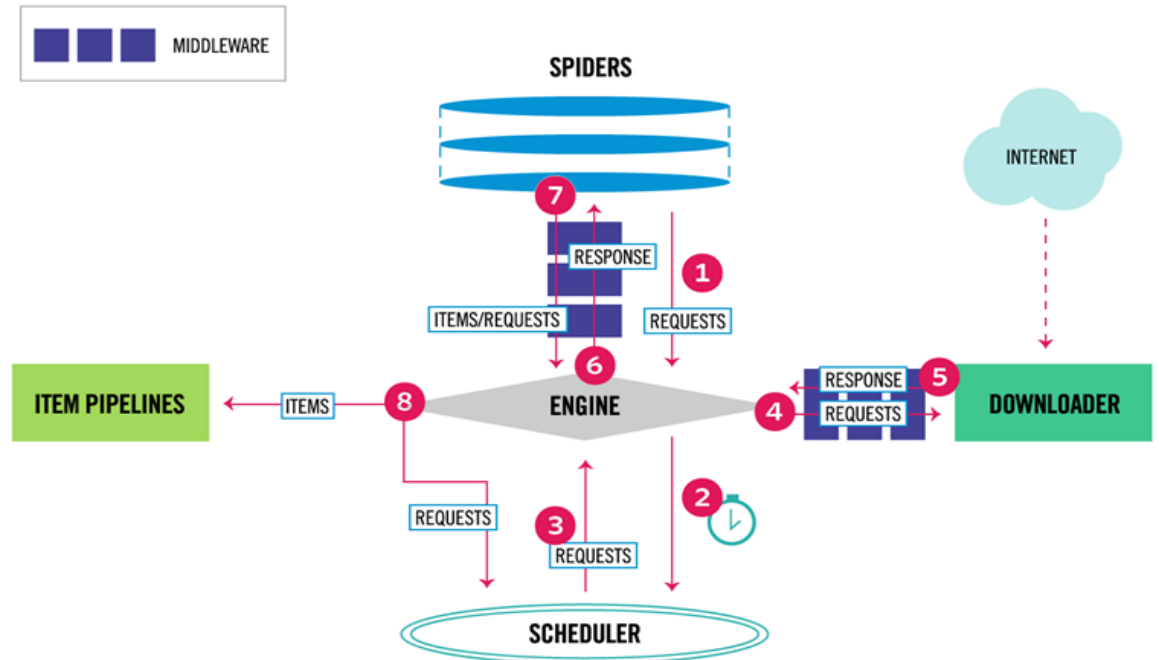
#### 1. Tổng quan hệ thống

Hệ thống lấy tin tự động sử dụng các công nghệ, dịch vụ, thư viện hỗ trợ cho quá trình thiết kế, cài đặt và vận hành bao gồm : Scrapyd, Nodejs Express, MongoDB, thư viện ElementTree.

- Quá trình lấy tin chủ yếu sử dụng trình đọc HTML (Scrapyd). Scrapyd là một dịch vụ hỗ trợ cực mạnh mẽ cho thư viện scrapy của python trong việc tạo và điều khiển các con spiders. Nó cho phép triển khai các dự án scrapy và điều khiển các con spiders của dự án đó bằng cách sử dụng HTTP JSON API.
- Nodejs Express framework để xây dựng một web services hỗ trợ cho việc lưu trữ và kiểm tra các liên kết URL được trích xuất từ RSS bởi scrapy.
- MongoDB là một cơ sở dữ liệu dạng NoSql được dùng để lưu trữ các liên kết URL.
- Thư viện ElementTree – The ElementTree XML API là thư viện dùng để trích xuất các liên kết URL trong RSS.

## 2. Kiến trúc hệ thống

### 2.1 Kiến trúc Scrapyd



Hình 3 Kiến trúc Scrapy

(nguồn: <https://doc.scrapy.org/en/latest/topics/architecture.html>)

Các thành phần trong sơ đồ:

- Scheduler: bộ lập lịch thứ tự các URL download.
- Downloader: thực hiện tải dữ liệu. Quản lý các lỗi khi download.
- Spiders: bóc tách dữ liệu thành các items và requests
- Item Pipeline: xử lý dữ liệu bóc tách được và lưu vào cơ sở dữ liệu.
- Scrapy Engine: quản lý các thành phần trên.

Luồng sự kiện	
Các sự kiện	Mô tả
Bước 1 (Sự kiện 1 – 2)	Cung cấp URL xuất phát (start_url), được tạo thành một Request lưu trong Scheduler.
Bước 2 (Sự kiện 3 – 4)	Scheduler lần lượt lấy các Requests gửi đến Downloader.
Bước 3 (Sự kiện 5 – 6)	Downloader tải dữ liệu từ internet, được Responses gửi đến Spiders.
Bước 4 (Sự kiện 7 – 8)	<ul style="list-style-type: none"> <li>Spider bóc tách dữ liệu, thu được các Item, gửi đến Item Pipeline.</li> <li>Spider tạo các Requests mới gửi đến Scheduler.</li> </ul>
Bước 5 (Sự kiện 8)	Item Pipeline thực hiện xử lý dữ liệu bóc tách được. Đơn giản nhất là thực hiện lưu dữ liệu vào database.
Bước 6	Kiểm tra Scheduler còn Request? <ul style="list-style-type: none"> <li>Đúng: Quay lại Bước 2.</li> <li>Sai: Kết thúc.</li> </ul>

*Bảng 1 Mô tả kiến trúc spaCy*

## 2.2 Hệ thống Web services Nodejs Express

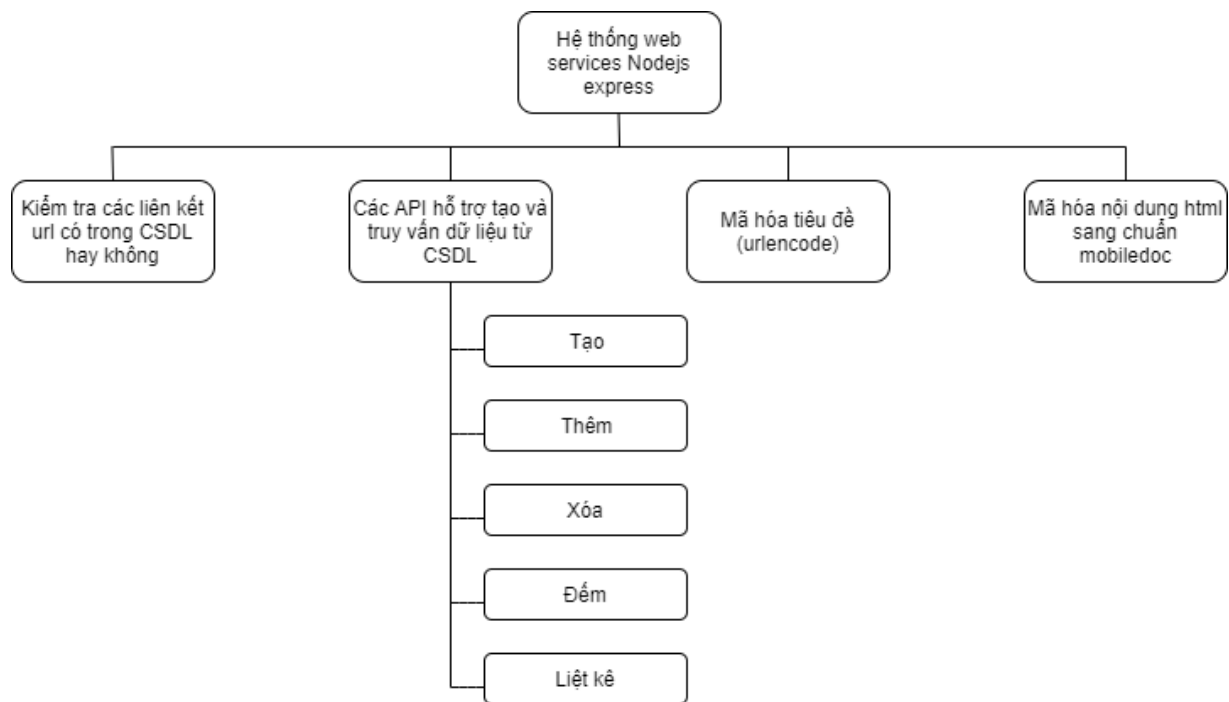
Để hỗ trợ cho hệ thống lấy tin và phân loại tin tự động, chúng tôi đã xây dựng một web services bằng Nodejs express, chứa các dịch vụ cung cấp các API hỗ trợ tích cực cho quá trình hoạt động của hệ thống lớn.

Trong hệ thống Nodejs Express, chúng tôi chủ yếu sử dụng hai phương thức của giao thức HTTP là GET và POST.

GET: Gửi thông tin người dùng đã được mã hóa thêm vào trên yêu cầu trang. GET lộ thông tin trên đường dẫn URL. Bằng thông của nó chỉ khoảng 1024 kí tự vì vậy GET hạn chế về số kí tự được gửi đi. GET không thể gửi dữ liệu nhị phân, hình ảnh ... Có thể được lưu vào bộ đệm và được đánh dấu trên trình duyệt. Lưu trong lịch sử trình duyệt.

POST: Phương thức POST truyền thông tin thông qua HTTP header, thông tin này được mã hóa như phương thức GET. Dữ liệu được gửi bởi phương thức POST rất bảo mật vì dữ liệu được gửi ngầm, không đưa lên URL, bằng việc sử dụng HTTPS, bạn có thể chắc chắn rằng thông tin của mình là an toàn. Các tham số được truyền trong nội dung của request nên có thể truyền dữ liệu lớn, hạn chế tùy thuộc vào cấu hình của Server. Không lưu vào bộ đệm và không đánh dấu trên trình duyệt, cũng như không được lưu lại trong lịch sử trình duyệt.

Sơ đồ chức năng của hệ thống web services Nodejs express



Hình 4 Sơ đồ chức năng hệ thống Nodejs

Các API được thiết kế trong web services

API POST parse : có chức năng chuyển đổi chuẩn HTML sang mobiledoc

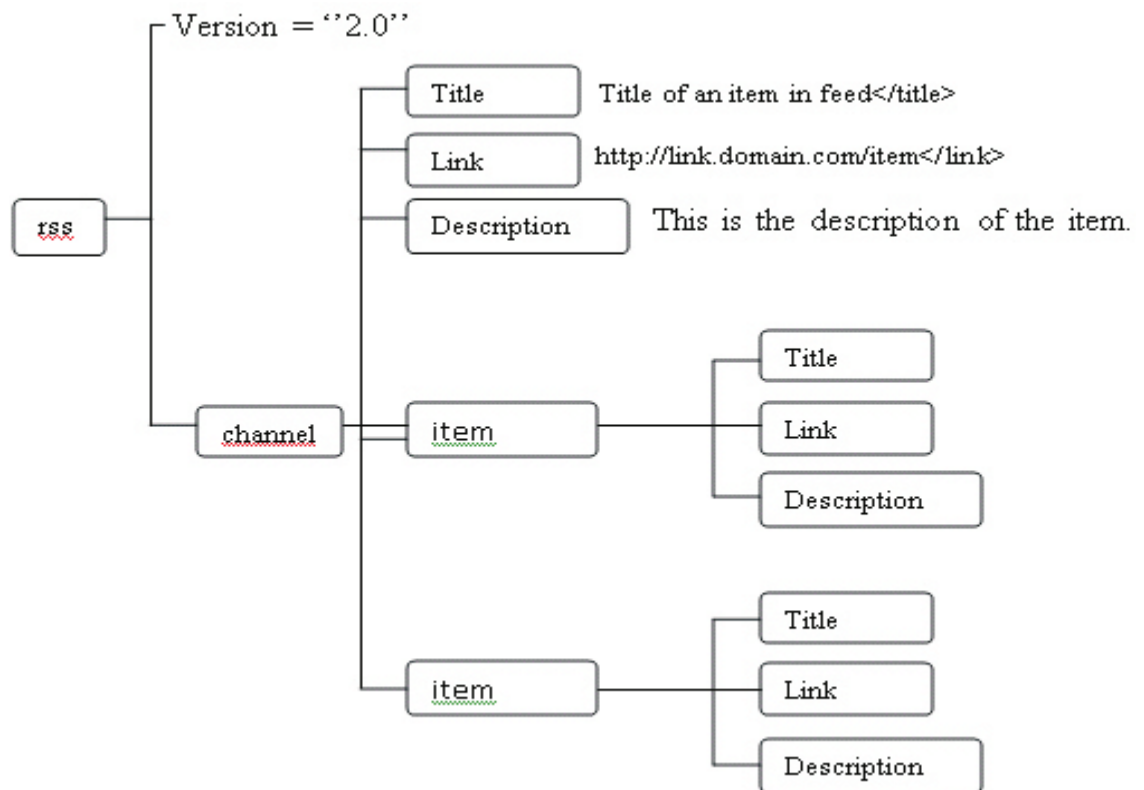
API POST check : có chức năng kiểm tra các bài tin từng của các trang tin, mỗi trang tin sẽ có một api check riêng.

API GET findall : liệt kê toàn bộ các liên kết URL có trong cơ sở dữ liệu, hỗ trợ quá trình kiểm tra, đánh giá hoạt động của hệ thống.

API GET drop : api hỗ trợ xóa cơ sở dữ liệu, phục vụ quá trình hoàn thiện hệ thống.

API POST spacy : api hỗ trợ quá trình phân loại tin tức.

## 2.3 Cấu trúc tập tin RSS



Hình 5 Cấu trúc tập tin RSS

(nguồn: [https://www.researchgate.net/figure/Structure-of-a-Simple-RSS-Document-Source\\_fig1\\_256935278](https://www.researchgate.net/figure/Structure-of-a-Simple-RSS-Document-Source_fig1_256935278))

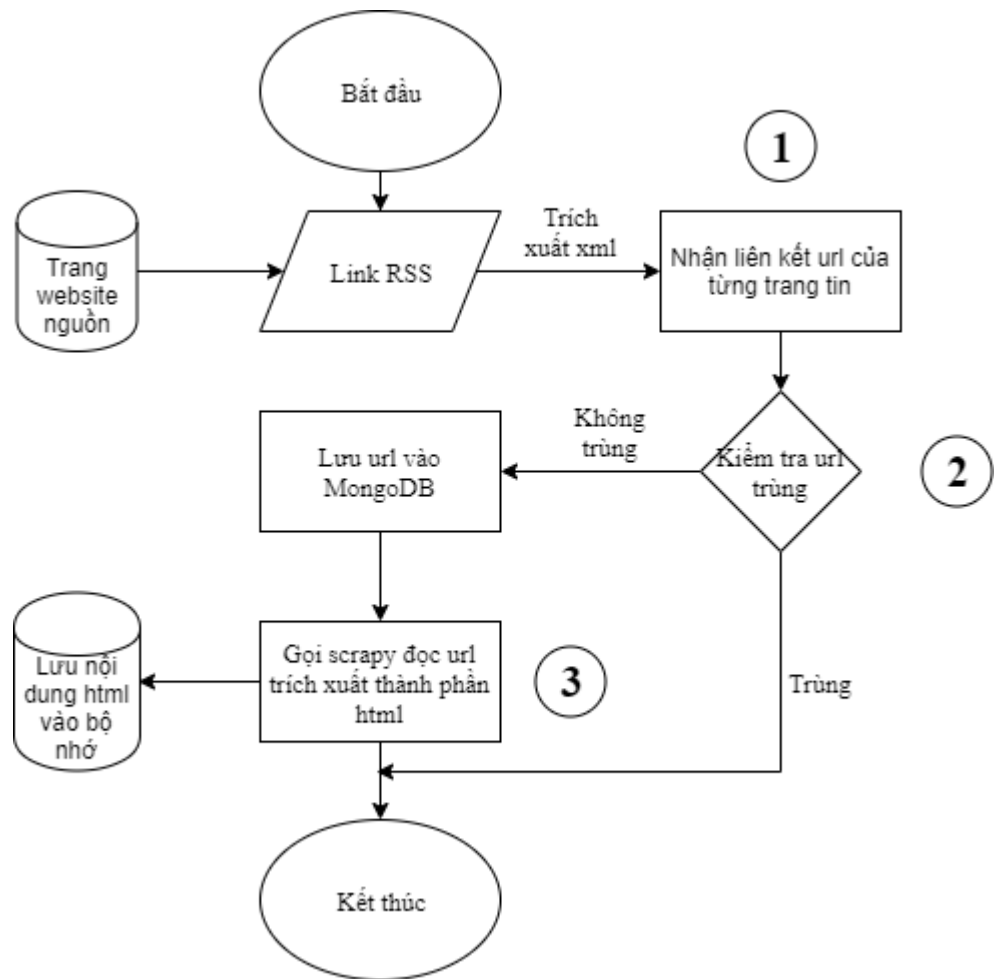
Một tập tin RSS bao gồm các phần :

Thẻ <rss version> : Cho biết phiên bản của tập tin RSS hiện tại.

Thẻ <channel> : Chứa các thành phần chính của trang RSS gồm các thẻ con Title, Link, Description và nhiều thẻ Item.

- Thẻ con <Title> : Chứa tên của trang RSS hiện tại.
- Thẻ con <Link> : Chứa liên kết đến trang chủ của website nguồn.
- Thẻ con <Description> : Mô tả thông tin trang RSS hiện tại.
- Thẻ con <Item> : Chứa các thông tin về liên kết URL đến những bài báo cần lấy dữ liệu. Mỗi thẻ Item lại có những thẻ con là Title, Link, Description.
  - Title : Chứa tiêu đề của bài tin
  - Link : Chứa liên kết URL dẫn đến bài tin
  - Description : Chứa nội dung chính của bài tin

## 2.4 Sơ đồ hoạt động hệ thống lấy tin



Hình 6 Sơ đồ hoạt động của quá trình lấy tin

Luồng sự kiện	
Các sự kiện	Mô tả
Sự kiện 1	Trích xuất các liên kết HTML từ tập tin RSS của trang web nguồn bằng thư viện ElementTree của python.
Sự kiện 2	Các liên kết URL sẽ đi qua server nodejs express để kiểm tra các liên kết có bị trùng lặp hay không. Nếu liên kết trùng sẽ bị loại bỏ, liên kết không trùng sẽ được lưu vào cơ sở dữ liệu.
Sự kiện 3	Scrapy sẽ gọi các liên kết được kiểm tra không trùng để trích xuất dữ liệu từ các URL đó, dữ liệu sau khi trích xuất sẽ được lưu vào bộ nhớ của hệ thống.

*Bảng 2 Mô tả quá trình lấy tin*

### 3. Thiết kế dữ liệu

#### 3.1 CSDL phi quan hệ - NoSql là gì ?

- NoSQL là 1 dạng CSDL mã nguồn mở và được viết tắt bởi: None-Relational SQL hay có nơi thường 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 1 mảnh vá cho những khuyết điểm và thiếu sót 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 bạn có thể mở rộng dữ liệu mà không lo tới những việc như tạo khóa ngoại, khóa chính, kiểm tra ràng buộc .v.v ...
- NoSQL bỏ qua tính toàn vẹn của dữ liệu và transaction để đổi lấy hiệu suất nhanh và khả năng mở rộng.

NoSQL được sử dụng ở rất nhiều công ty, tập đoàn lớn, ví dụ như FaceBook sử dụng Cassandra do FaceBook phát triển, Google phát triển và sử dụng BigTable,...

### 3.2 Cơ bản về MongoDB

MongoDB là một CSDL 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.

Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB.

### 3.3 Cấu trúc cơ sở dữ liệu sử dụng trong hệ thống

Trong hệ thống, MongoDB được sử dụng để lưu trữ các liên kết URL hỗ trợ cho việc kiểm tra các URL bị trùng lặp.

```
[
  {
    "_id": "5da12e9d5b1a5557e8931fa5",
    "url": "https://vietnambiz.vn/ba-tuyen-giao-thong-ket-noi-voi-san-bay-long-thanh-20191012074945158.htm"
  },
  {
    "_id": "5da12e9d5b1a556261931fa6",
    "url": "https://vietnambiz.vn/han-nhat-ket-thuc-vong-dam-phan-thu-nhat-ve-tranh-cai-thuong-mai-201910120747476.htm"
  },
  {
    "_id": "5da12e9d5b1a5531fe931fa7",
    "url": "https://vietnambiz.vn/aramco-van-tiep-tuc-ipo-sau-vu-tan-cong-hai-co-so-dau-mo-2019101207145161.htm"
  },
  {
    "_id": "5da12e9d5b1a556893931fa8",
    "url": "https://vietnambiz.vn/ty-gia-ngan-hang-acb-moi-nhat-ngay-1-10-2019-20191001160727495.htm"
  },
  {
    "_id": "5da12e9d5b1a55c953931fa9",
    "url": "https://vietnambiz.vn/ty-gia-ngan-hang-techcombank-moi-nhat-thang-10-2019-20191001171321277.htm"
  },
  {
    "_id": "5da12e9d5b1a557888931faa",
    "url": "https://vietnambiz.vn/ty-gia-ngan-hang-eximbank-moi-nhat-thang-10-20191001185953848.htm"
  },
]
```

*Hình 7 Cấu trúc dữ liệu trong MongoDB*

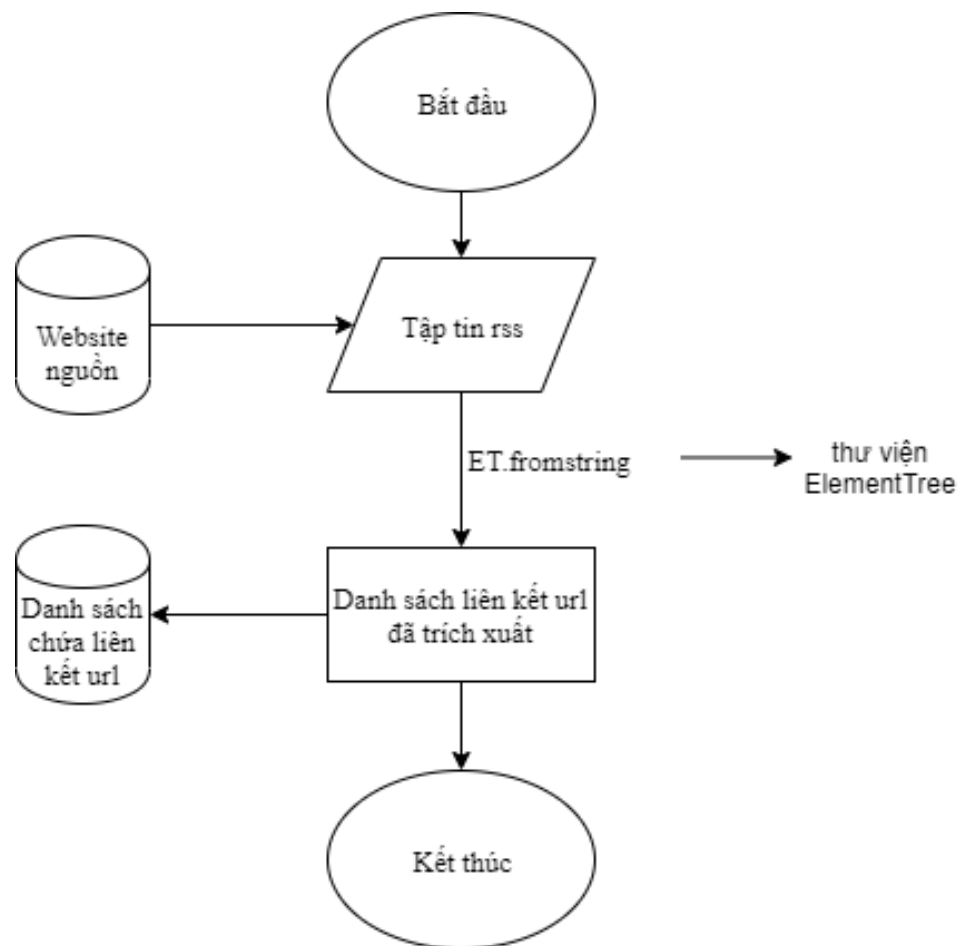


Định danh	Mô tả
_id	Mã liên kết
URL	Liên kết

Bảng 3 Mô tả thành phần cấu trúc dữ liệu

#### 4. Thiết kế hệ thống lấy tin tự động

##### 4.1 Quá trình trích xuất liên kết URL từ tập tin RSS (Bước 1)

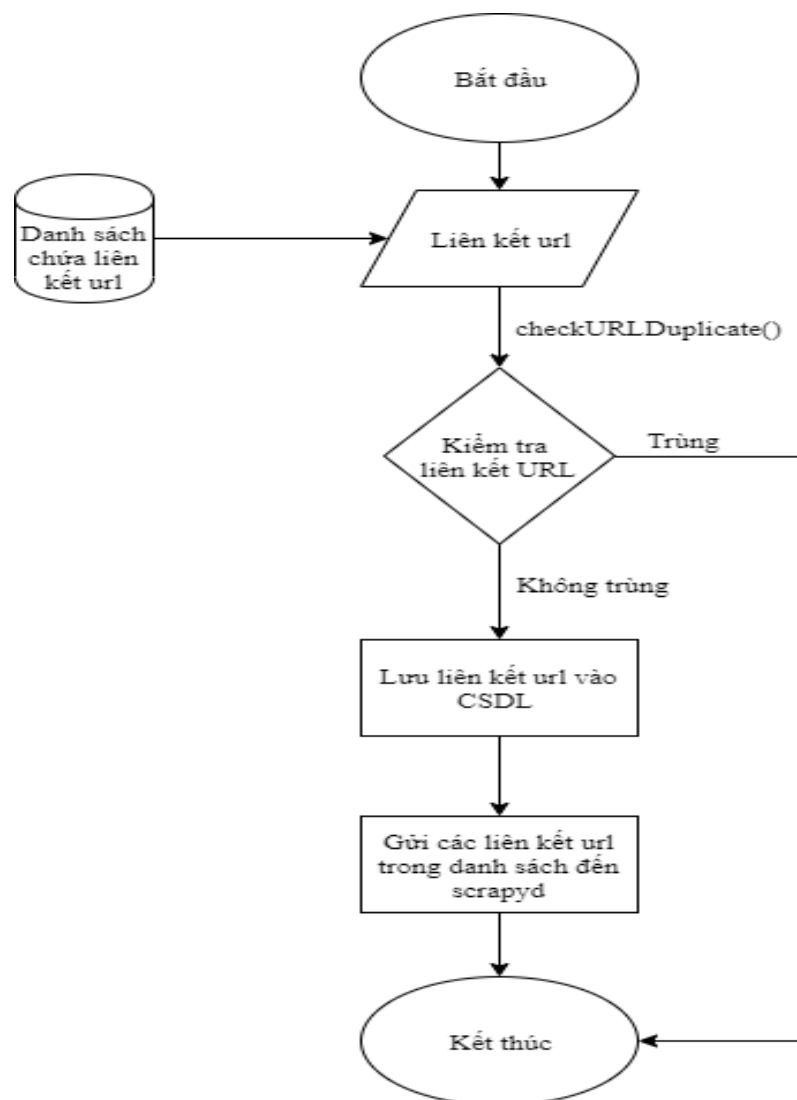


Hình 8 Sơ đồ hoạt động quá trình trích xuất liên kết URL

<b>Luồng sự kiện</b>	
<b>Các sự kiện</b>	<b>Mô tả</b>
Sự kiện 1	Nhận tập tin RSS từ đường dẫn của trang web nguồn
Sự kiện 2	Duyệt tập tin RSS bằng hàm ET.fromstring của thư viện ElementTree, tách được các thành phần của tập tin RSS, từ đó rút ra được liên kết URL của các trang tin tức.
Sự kiện 3	Lưu các liên kết URL trích xuất được thành một danh sách.

*Bảng 4 Mô tả quá trình trích xuất liên kết URL*

#### 4.2 Quá trình kiểm tra liên kết URL (Bước 2)



Hình 9 Sơ đồ quá trình kiểm tra liên kết URL

Luồng sự kiện	
Các sự kiện	Mô tả
Sự kiện 1	Hàm checkDuplicate sẽ gọi lần lượt các liên kết URL để kiểm tra.
Sự kiện 2	Sẽ có 2 trường hợp đối với liên kết URL: <ul style="list-style-type: none"><li>• Nếu trùng : Loại bỏ.</li><li>• Nếu không trùng : Liên kết URL sẽ được lưu vào danh sách và được gửi qua scrapyd.</li></ul>

*Bảng 5 Mô tả quá trình kiểm tra liên kết URL*

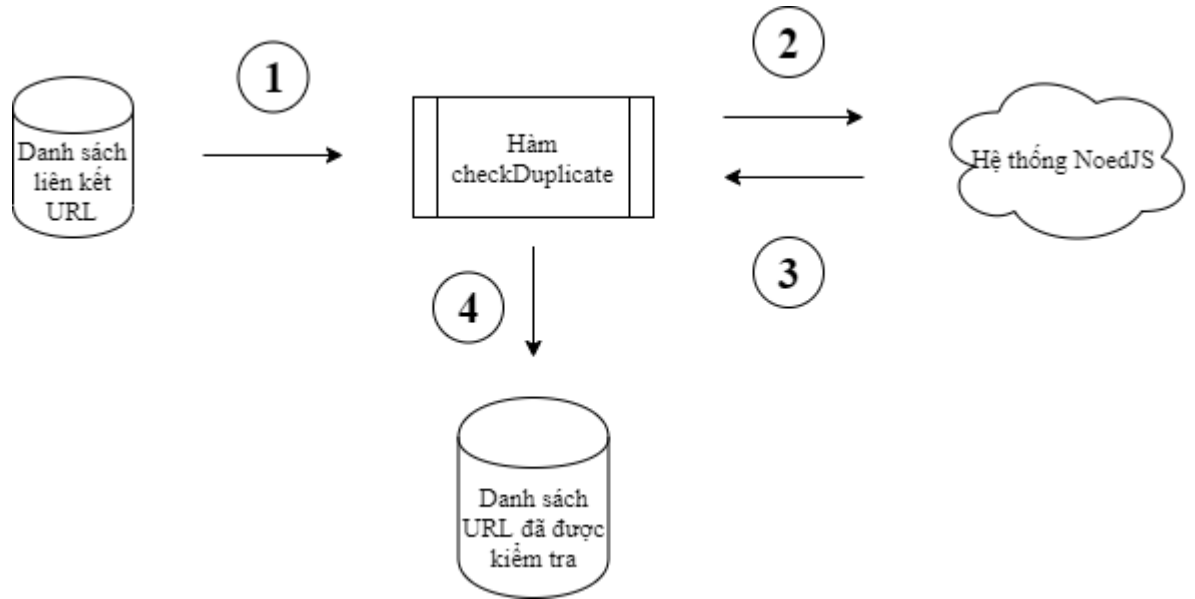
#### ❖ Mô tả về hàm checkDuplicate

Hàm checkDuplicate được khởi tạo nhằm mục đích kiểm tra các liên kết URL được trích xuất từ tập RSS có bị trùng hay không.

##### ✓ Tại sao cần phải kiểm tra các liên kết URL ?

Hệ thống lấy tin tự động vận hành dựa trên thời gian chạy được đặt lịch sẵn. Nên sẽ có những trường hợp các bài tin được lấy về trùng lặp với nhau. Ví dụ cụ thể như hệ thống được cài đặt cứ mỗi hai giờ sẽ tiến hành lấy tin từ trang VietnamBiz, thì khi trang không có bài báo mới, hệ thống lấy tin ở lần chạy thứ hai sẽ lấy toàn bộ các tin đã lấy ở lần chạy đầu tiên, tạo trường hợp trùng lặp dữ liệu, nếu tiếp tục xử lý các tin này sẽ gây nên sự dư thừa và làm chậm hệ thống, vì vậy cần kiểm tra các liên kết URL.

✓ Cách thức hoạt động của hàm checkDuplicate như thế nào ?

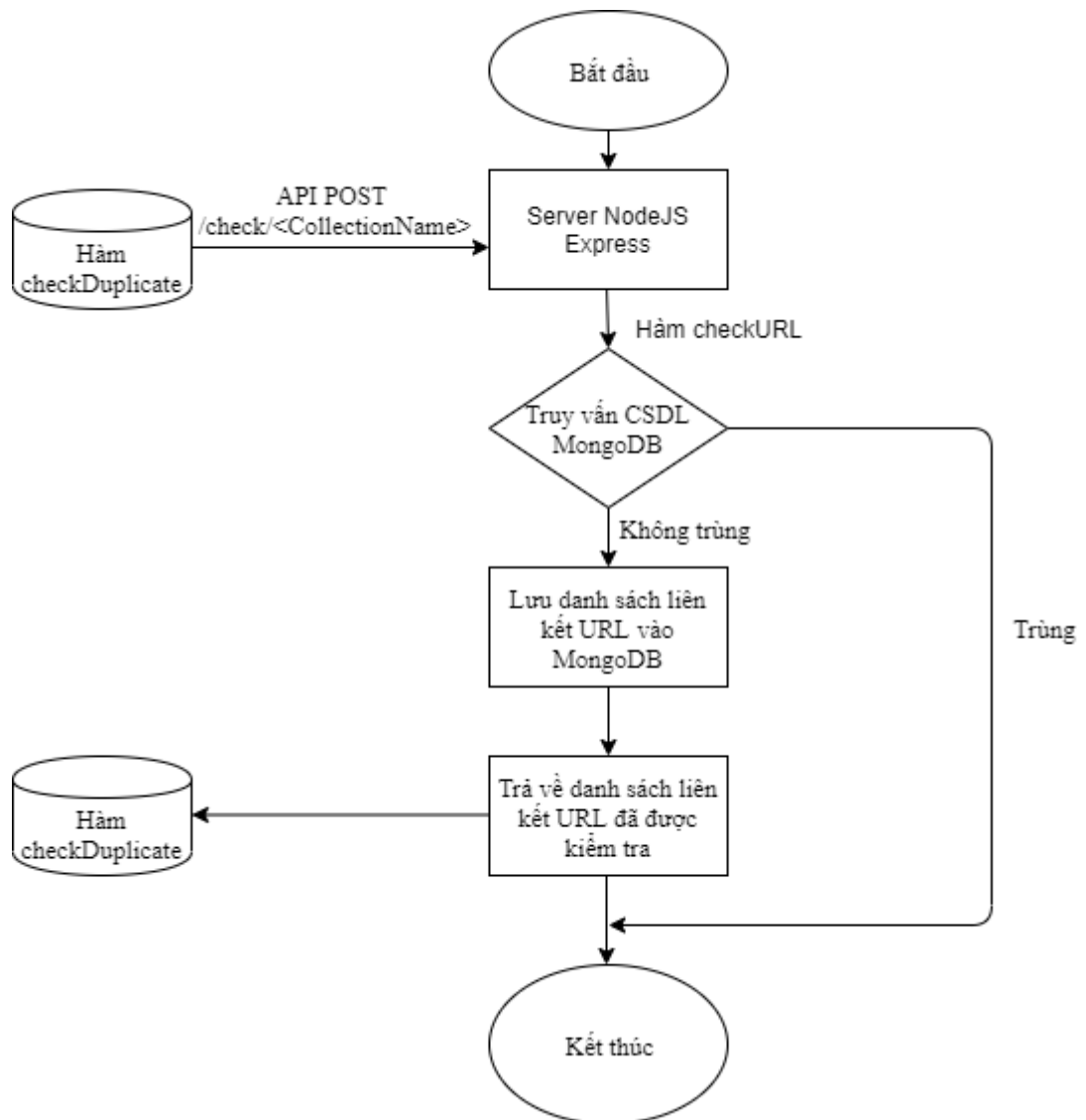


Hình 10 Sơ đồ hoạt động của hàm checkDuplicate

Luồng sự kiện	
Các sự kiện	Mô tả
Sự kiện 1	Hàm checkDuplicate nhận danh sách liên kết URL.
Sự kiện 2	Mã hóa danh sách URL thành dạng json và gửi đến hệ thống NodeJS thông qua API POST /check.
Sự kiện 3	Server NodeJS tiến hành xử lý dữ liệu và gửi trả danh sách kết quả liên kết URL đã được kiểm tra (các liên kết URL không nằm trong CSDL) được mã hóa theo JSON.
Sự kiện 4	Đọc tập tin JSON được trả về từ server NodeJS. Lưu kết quả trả về thành một danh sách các liên kết URL đã được kiểm tra.

Bảng 6 Mô tả quá trình hoạt động hàm checkDuplicate

#### 4.3 Quá trình kiểm tra liên kết URL của hệ thống web services NodeJS

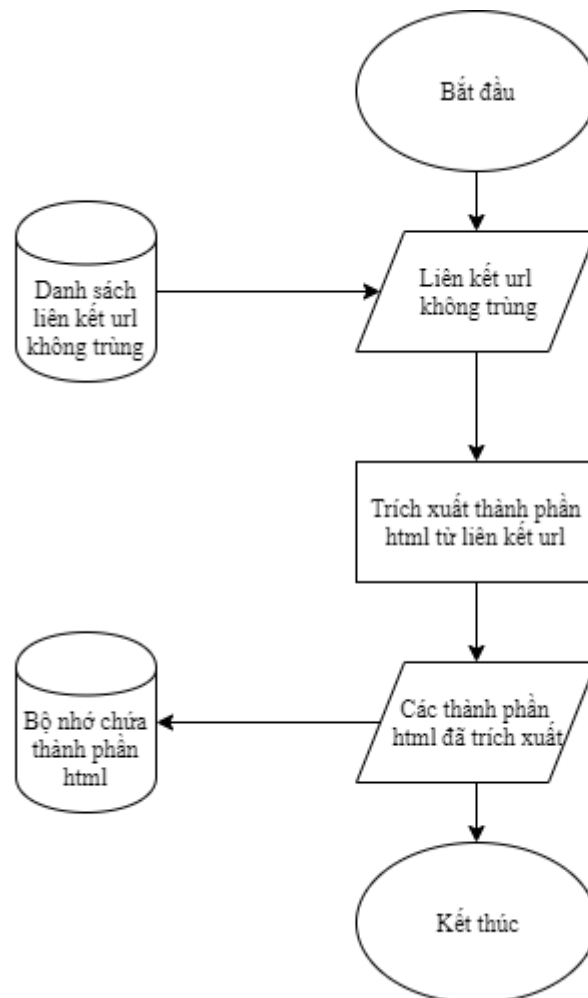


Hình 11 Sơ đồ quá trình kiểm tra liên kết URL của Nodejs

<b>Luồng sự kiện</b>	
<b>Các sự kiện</b>	<b>Mô tả</b>
Sự kiện 1	Server NodeS nhận danh sách các liên kết URL thông qua API POST /check/<CollectionName> từ hàm checkDuplicate
Sự kiện 2	Server NodeJS gọi hàm checkURL để tạo kết nối đến CSDL MongoDB theo <CollectionName> và kiểm tra danh sách liên kết URL
Sự kiện 3	<p>Hàm checkURL :</p> <ul style="list-style-type: none"> <li>• Tạo kết nối đến CSDL MongoDB theo &lt;CollectionName&gt;</li> <li>• Kiểm tra lần lượt các liên kết URL trong danh sách liên kết URL, bằng cách truy vấn CSDL</li> <li>• Các liên kết URL không nằm trong CSDL được thêm vào danh sách liên kết URL đã kiểm tra, bỏ qua các liên kết URL có trong CSDL.</li> <li>• Thêm danh sách liên kết URL đã được kiểm tra vào CSDL.</li> </ul>
Sự kiện 4	Trả về danh sách kết quả các liên kết URL đã kiểm tra

*Bảng 7 Mô tả quá trình kiểm tra liên kết URL của Nodejs server*

#### 4.4 Quá trình Scrapyd đọc liên kết URL và trích xuất HTML (Bước 3)



Hình 12 Sơ đồ quá trình trích xuất HTML của Scrapy



Luồng sự kiện	
Các sự kiện	Mô tả
Sự kiện 1	Lấy các liên kết URL từ danh sách liên kết URL không trùng
Sự kiện 2	<p>Gọi scrapyd xử lý liên kết URL trích các thành phần HTML trong URL đó ra</p> <p>Các thành phần HTML cần cho hệ thống gồm:</p> <ul style="list-style-type: none"><li>• Thẻ chứa tiêu đề</li><li>• Thẻ chứa nội dung chính</li><li>• Thẻ chứa toàn bộ nội dung bài viết</li><li>• Liên kết bài viết</li><li>• Thẻ chứa tác giả bài viết</li></ul>
Sự kiện 3	Lưu các thành phần HTML trích xuất được vào bộ nhớ.

*Bảng 8 Mô tả quá trình đọc liên kết URL và trích xuất HTML*

## **II. PHÂN HỆ II : XÂY DỰNG HỆ THỐNG PHÂN LOẠI TIN TỰ ĐỘNG**

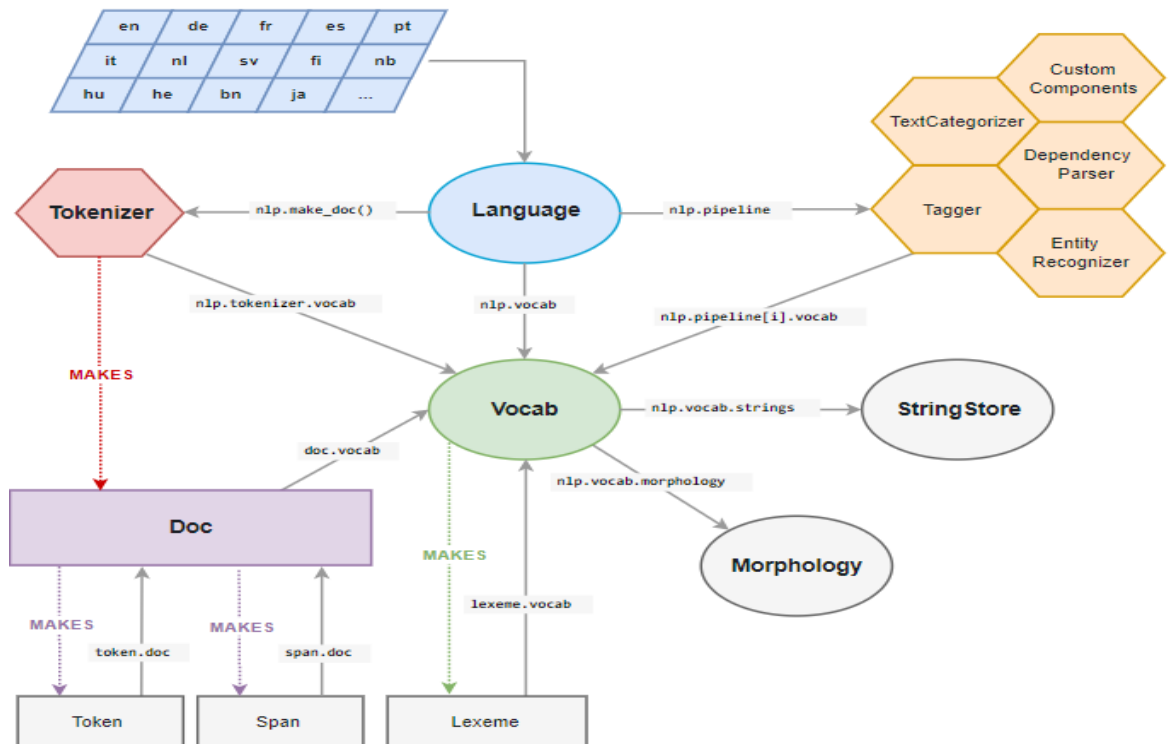
### **1. Tổng quan hệ thống**

Hệ thống phân loại tin tự động sử dụng các công nghệ, dịch vụ, thư viện hỗ trợ quá trình cài đặt và vận hành bao gồm: Nodejs express server, thư viện Flask, thư viện urllib.parse, thư viện spaCy.

- Nodejs server sử dụng Express framework để xây dựng một server hỗ trợ cho việc lưu trữ và kiểm tra loại (tag) của bài viết sau mỗi lần huấn luyện.
- Flask là một thư viện hỗ trợ tạo server Python cho hệ thống trong quá trình tạo tập huấn luyện và phân loại.
- Urllib.parse là thư viện hỗ trợ chuyển đổi dạng của dữ liệu phù hợp với yêu cầu của hệ thống.
- SpaCy là thư viện chính của toàn bộ hệ thống phân loại, spaCy hỗ trợ từ việc tạo tập huấn luyện, tập kiểm tra, hơn thế nữa, spaCy còn hỗ trợ tạo ra model chuẩn cho hệ thống hoạt động ổn định sau khi tập kiểm tra cho kết quả mong muốn.

## 2. Kiến trúc hệ thống

### 2.1 Kiến trúc của SpaCy



Hình 13 Kiến trúc spaCy

(nguồn: <https://spacy.io/api>)

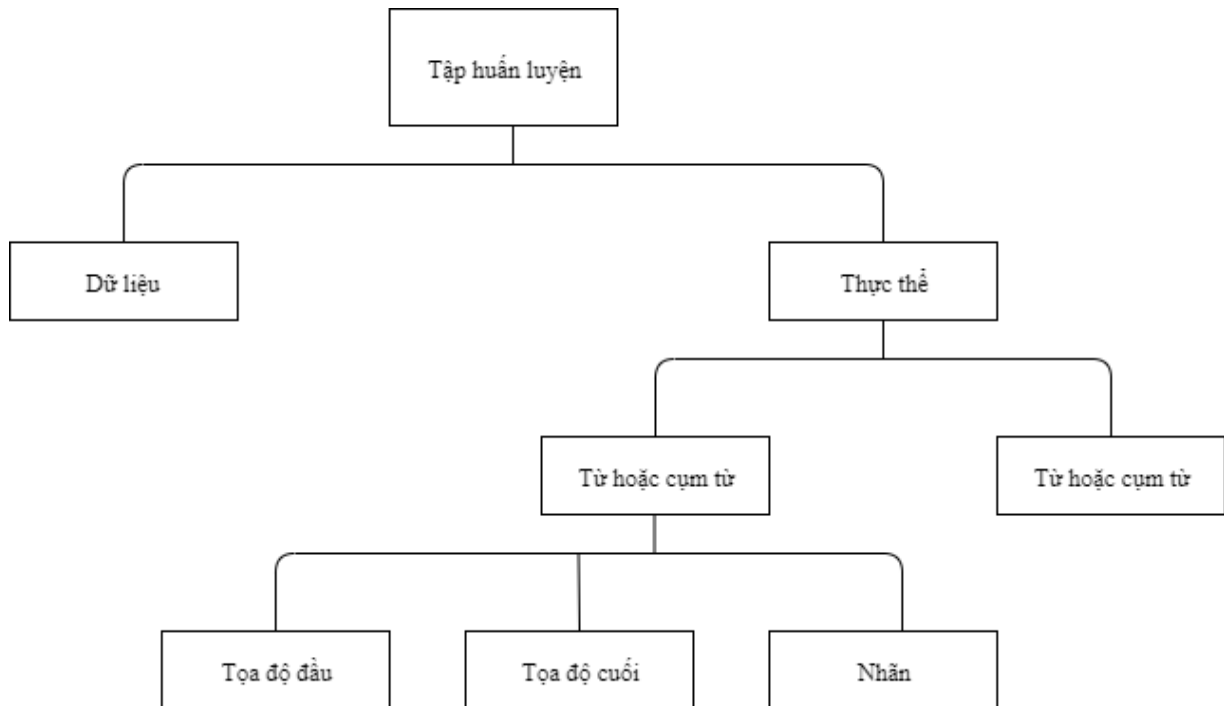
Các thành phần trong kiến trúc spaCy	
Tên	Mô tả
Doc	Một thành phần để truy cập các chú thích ngôn ngữ
Span	Một cụm từ được cắt từ Doc.
Token	Một mã thông báo riêng lẻ - nghĩa là một từ, ký hiệu dấu chấm câu, khoảng trắng.
Lexeme	Một mục trong từ vựng. Nó có một loại từ không có ngữ cảnh, trái ngược với Token.

Language	Một quá trình xử lý văn bản. Thông thường, bạn sẽ tải cái này một lần cho mỗi tiến trình dưới dạng nlp và truyền ví dụ xung quanh ứng dụng của bạn.
Tokenizer	Phân đoạn văn bản và tạo đối tượng Doc các đối tượng với các cụm từ đã được cắt.
Lemmatizer	Xác định các dạng cơ sở của từ
Morphology	Gán các tính năng ngôn ngữ như bỏ đề, trường hợp danh từ, thì của động từ, vv dựa trên từ và thể một phần của bài phát biểu.
Tagger	Chú thích các trên các đối tượng Doc.
DependencyParser	Chú thích phụ thuộc cú pháp vào các đối tượng Doc.
EntityRecognizer	Chú thích các thực thể được đặt tên riêng, ví dụ: người hoặc sản phẩm, trên các đối tượng Doc.
TextCategorizer	Gán danh mục hoặc nhãn cho các đối tượng Doc.
Matcher	Kết hợp các chuỗi mã thông báo, dựa trên quy tắc mẫu, tương tự như biểu thức thông thường.
PhraseMatcher	Kết hợp các chuỗi mã thông báo dựa trên các cụm từ
EntityRuler	Thêm các khoảng thực thể vào Tài liệu bằng cách sử dụng quy tắc dựa trên mã thông báo hoặc kết hợp cụm từ chính xác.
Sentencizer	Thực hiện logic phát hiện ranh giới câu tùy chỉnh mà không cần phải phân tích cú pháp phụ thuộc.
Other functions	Tự động áp dụng một cái gì đó cho Tài liệu, ví dụ: để hợp nhất các chuỗi mã thông báo.
Vocab	Một bảng tra cứu từ vựng cho phép bạn truy cập các đối tượng Lexeme.

StringStore	Ánh xạ chuỗi đến và từ các giá trị băm.
Vectors	Lớp thành phần cho dữ liệu vector được khóa bởi chuỗi.
GoldParse	Bộ sưu tập cho các chú thích đào tạo.
GoldCorpus	Một kho văn bản có chú thích, sử dụng định dạng tệp JSON. Quản lý các chú thích để gắn thẻ, phân tích cú pháp phụ thuộc và NER (Name Entity Recognizer).

Bảng 9 Các thành phần trong kiến trúc spaCy

## 2.2 Cấu trúc tập huấn luyện (tập train)



Hình 14 Cấu trúc tập huấn luyện

Ví dụ cụ thể:

```
[("thám hiểm là du lịch.", {"entities" : [(0, 9, "Du Lịch"), (13, 20, "Du Lịch")]}),
```

Trong ảnh trên ta có thể thấy các thành phần của cấu trúc tập huấn luyện như sau:

Dữ liệu: thám hiểm là du lịch

Thực thể: entities

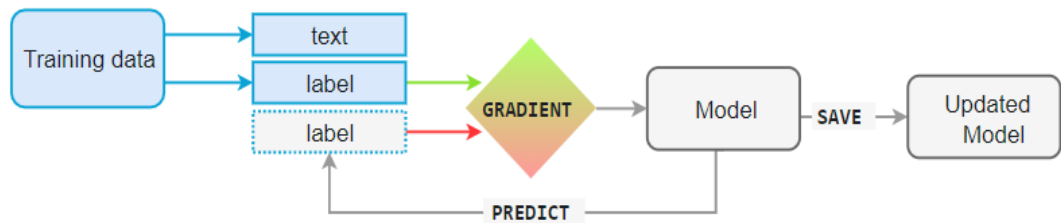
Từ hoặc cụm từ: là các từ hoặc cụm từ trong dữ liệu được tiên đoán nhãn.

Tọa độ đầu: vị trí chữ đầu tiên của từ hoặc cụm từ được tiên đoán nhãn (khoảng trắng cũng được tính).

Tọa độ cuối: Vị trí kết thúc của từ hoặc cụm từ được tiên đoán nhãn.

Nhãn: nhãn được tiên đoán cho từ hoặc cụm từ trong dữ liệu.

### 2.3 Sơ đồ quá trình huấn luyện của spaCy



Hình 15 Kiến trúc quá trình đào tạo tập huấn luyện của spaCy

(nguồn: <https://spacy.io/usage/training>)

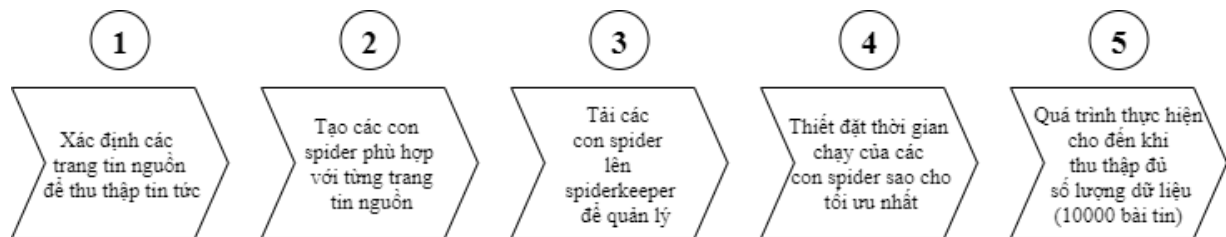
Luồng sự kiện	
Sự kiện	Mô tả
Sự kiện 1	Xác định thành phần của dữ liệu bao gồm “chuỗi” (text) và “nhãn” (label).
Sự kiện 2	Chuyển “chuỗi” và “nhãn” qua hàm Gradient.

Sự kiện 3	“Chuỗi” và “nhãn” sau khi đi qua hàm Gradient sẽ cho ra một “mẫu” (model).
Sự kiện 4	Khi có một dữ liệu mới đi qua “mẫu” (model). “Mẫu” sẽ tiến hành dự đoán (predict) “nhãn” của dữ liệu mới để tạo ra cặp “chuỗi” và “nhãn” mới.
Sự kiện 5	Tiến hành cập nhật lại “mẫu” cho đến khi không còn dữ liệu đầu vào.

*Bảng 10 Mô tả quá trình huấn luyện của spaCy*

- ❖ Gradient: Trọng số của hàm mất tính toán (loss function) sự khác biệt giữa đầu vào và đầu ra dự kiến.

## 2.4 Quá trình thu thập dữ liệu



*Hình 16 Quá trình thu thập dữ liệu*

**Bước 1:** Cần phải xác định được các trang tin nguồn nào có hỗ trợ xem tin với định dạng RSS.

**Bước 2:** Viết chương trình tạo ra các con spider phù hợp với cấu trúc của bài tin đối với từng trang tin nguồn.

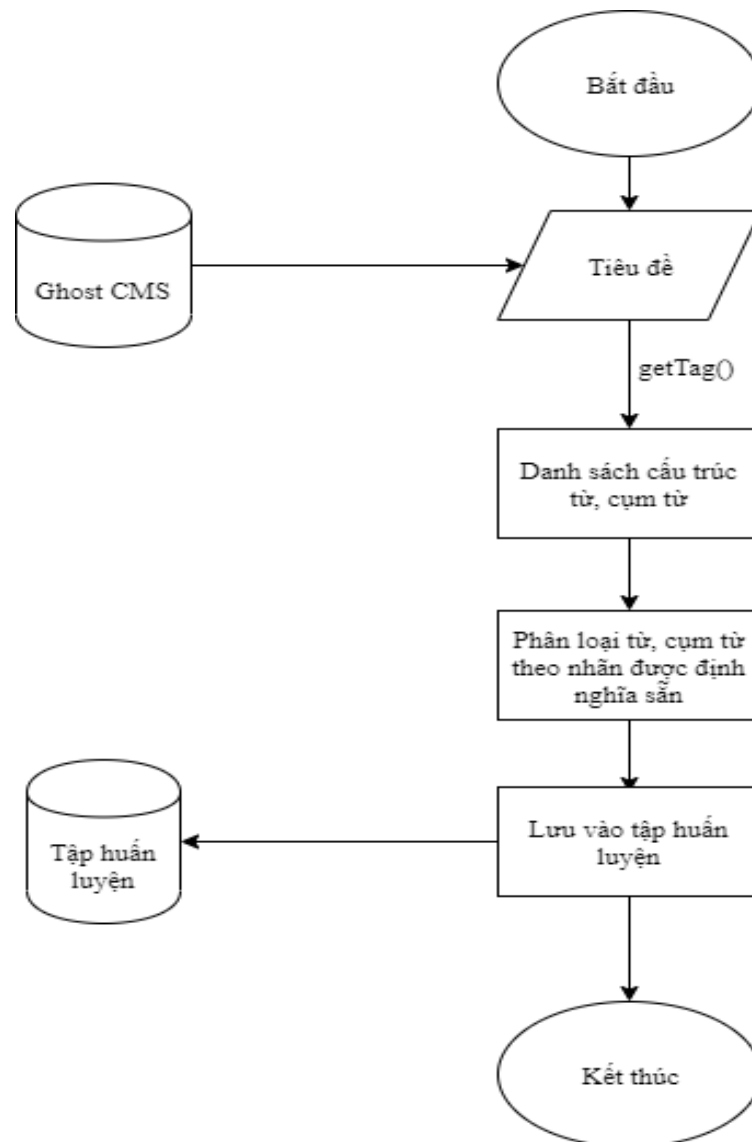
**Bước 3:** Tiến hành tải các con spider lên hệ thống Spiderkeeper sau khi các con spider đã hoạt động được.

**Bước 4:** Lựa chọn thời gian chạy của từng con spider sao cho tin tức lấy về từ các trang tin nguồn là tối ưu nhất về mặt hạn chế tin trùng lặp.

**Bước 5:** Quá trình sẽ thực hiện liên tục cho đến khi thu thập đủ số lượng tin cần thiết để tạo ra tập huấn luyện.

## 2.5 Quá trình tạo tập huấn luyện (tập train) cho SpaCy

Tin tức là một đề tài vô cùng đa dạng về chữ nghĩa cũng như vốn từ ngữ cực kỳ phức tạp, vì thế cần phải có một tập huấn luyện đủ lớn để có thể đưa ra những tên loại phù hợp với bài báo ở mức độ chính xác có thể chấp nhận được. Trong hệ thống này, chúng tôi quyết định chọn tập huấn luyện có kích thước mười nghìn bài báo từ tất cả những trang tin tức nguồn. Từ mười nghìn bài báo này chúng tôi sẽ huấn luyện ra một “mẫu” (model) để ứng dụng vào hệ thống.



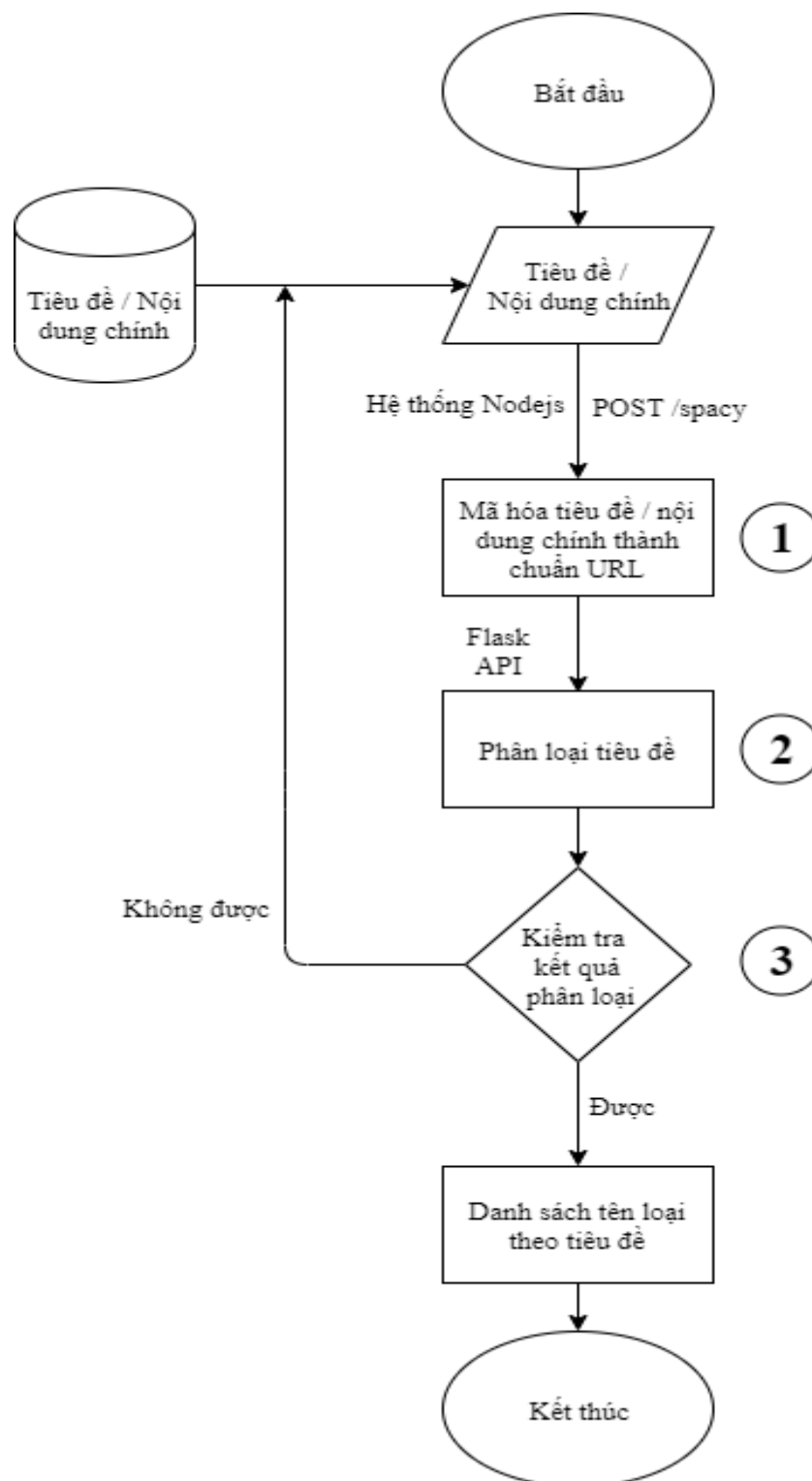
Hình 17 Sơ đồ hoạt động quá trình tạo tập huấn luyện



<b>Luồng sự kiện</b>	
<b>Các sự kiện</b>	<b>Mô tả</b>
Sự kiện 1	Lấy tiêu đề từ Ghost CMS thông qua API GET của Ghost.
Sự kiện 2	Gọi hàm getTag() tách tiêu đề thành từ và cụm từ.
Sự kiện 3	Phân loại từ và cụm từ theo mẫu được định nghĩa sẵn.
Sự kiện 4	Lưu danh sách cấu trúc các từ, cụm từ được định nghĩa và nhãn tương ứng vào tập huấn luyện.

*Bảng 11 Mô tả quá trình tạo tập huấn luyện*

## 2.6 Sơ đồ hoạt động hệ thống phân loại tin



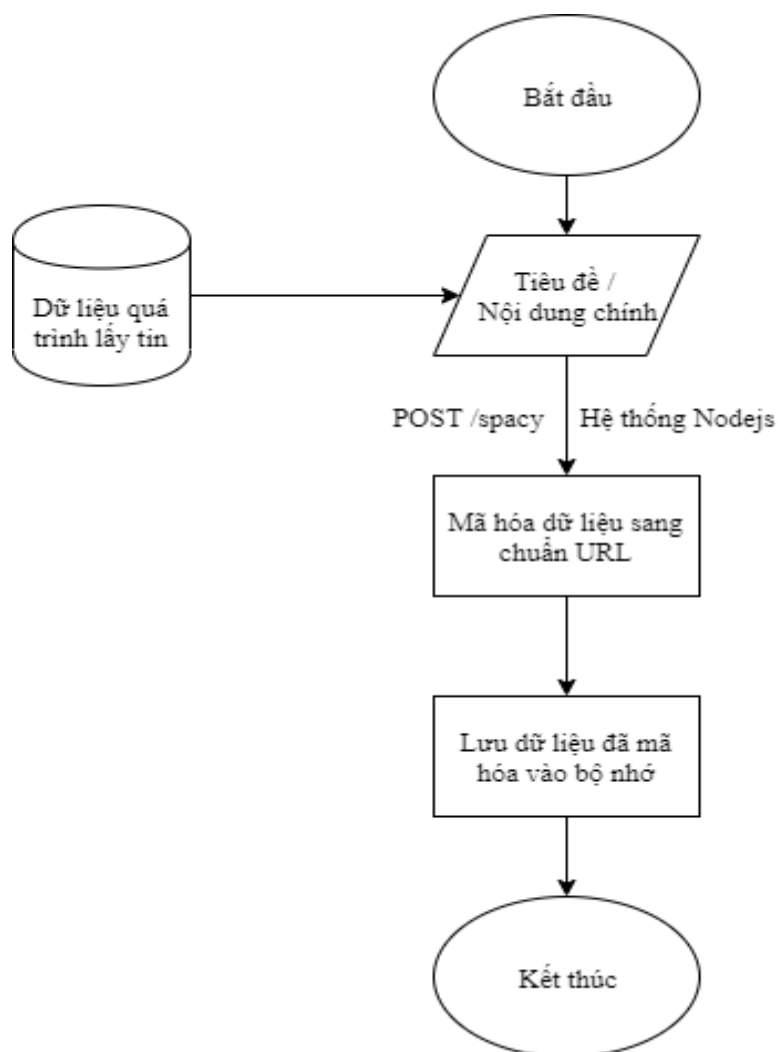
Hình 18 Sơ đồ hoạt động của hệ thống phân loại tin

<b>Luồng sự kiện</b>	
<b>Các sự kiện</b>	<b>Mô tả</b>
Sự kiện 1	Chọn tiêu đề làm nội dung để phân loại.
Sự kiện 2	Mã hóa tiêu đề theo chuẩn URL thông qua hệ thống Nodejs bằng cách gọi API POST /spaCy.
Sự kiện 3	Gửi tiêu đề đã mã hóa đến Flask API để phân loại.
Sự kiện 4	<p>Phân loại:</p> <p>Trường hợp 1: Tiêu đề được phân loại thành công theo nhãn thông qua “mẫu”. Lưu danh sách nhãn của tiêu đề.</p> <p>Trường hợp 2: Tiêu đề phân loại không thành công (không xác định được nhãn của tiêu đề). Quay lại sự kiện 1 và chọn “nội dung chính” làm nội dung để phân loại.</p>

*Bảng 12 Mô tả quá trình phân loại tin*

### 3. Thiết kế hệ thống phân loại tin

#### 3.1 Quá trình mã hóa tiêu đề / nội dung chính sang chuẩn URL (Bước 1)

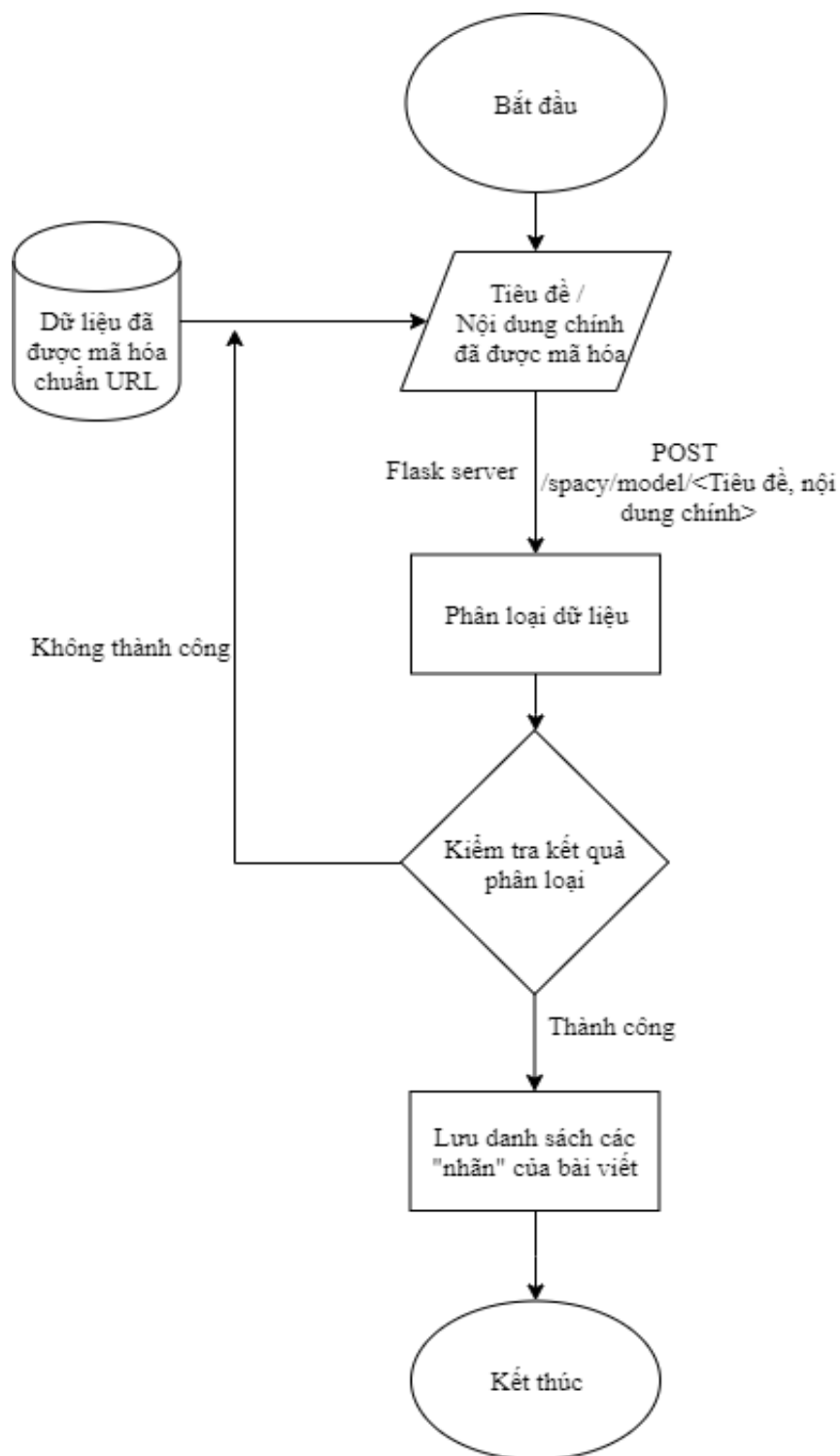


Hình 19 Quá trình mã hóa dữ liệu sang chuẩn URL

Luồng sự kiện	
Các sự kiện	Mô tả
Sự kiện 1	Gửi dữ liệu lấy được qua hệ thống Nodejs thông qua API POST /spacy
Sự kiện 2	Thực hiện mã hóa dữ liệu sang chuẩn URL bằng thư viện urlencode.
Sự kiện 3	Lưu dữ liệu đã được mã hóa vào bộ nhớ.

*Bảng 13 Mô tả quá trình mã hóa dữ liệu sang URL*

### 3.2 Quá trình phân loại tin tức của Flask server (Bước 2 và 3)

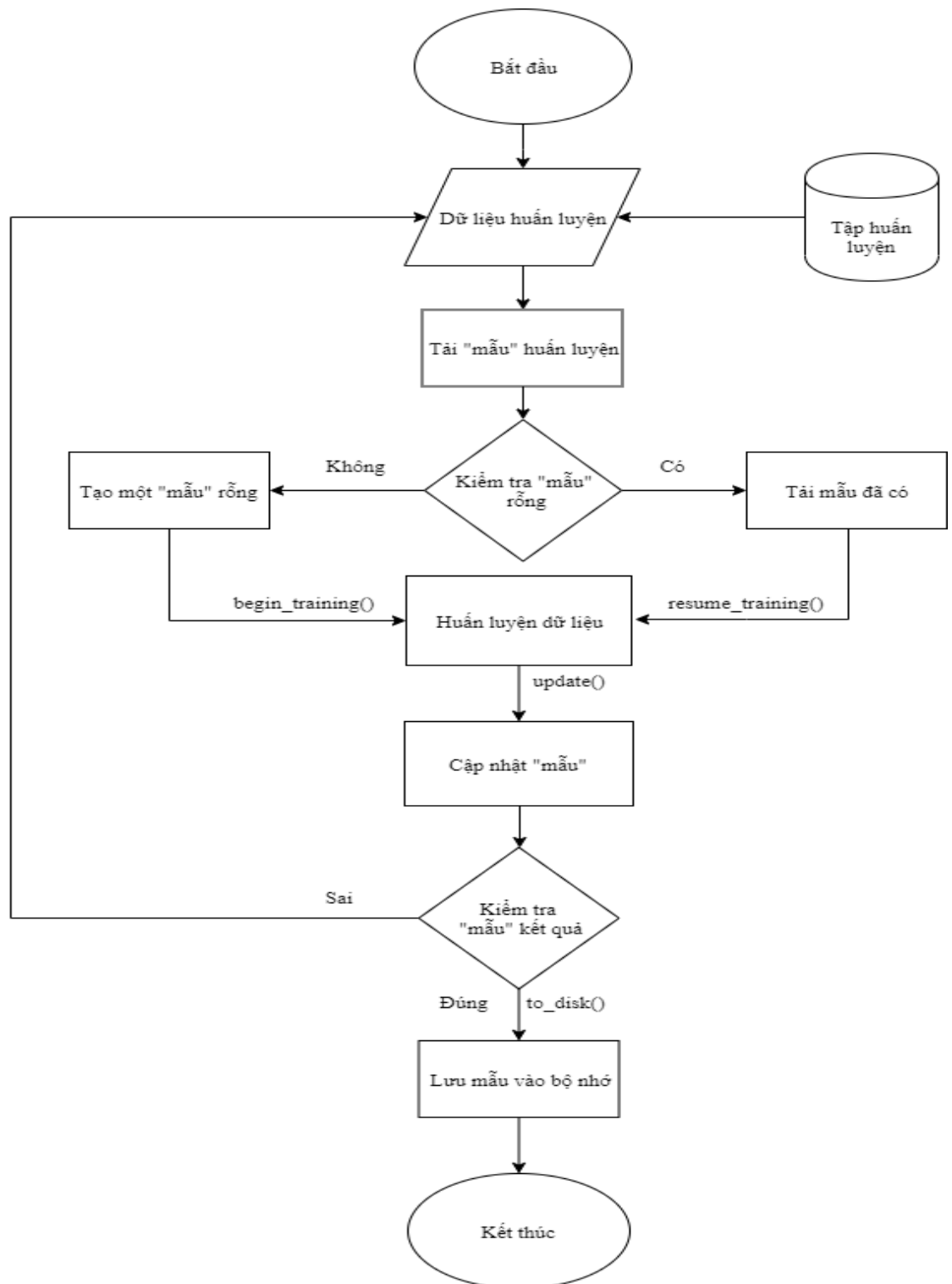


Hình 20 Quá trình phân loại tin tức của Flask

<b>Luồng sự kiện</b>	
<b>Các sự kiện</b>	<b>Mô tả</b>
Sự kiện 1	Nhận dữ liệu là tiêu đề đã được mã hóa từ quá trình mã hóa URL.
Sự kiện 2	Gửi dữ liệu sang Flask server để tiến hành phân loại thông qua API POST /spacy/model/<dữ liệu>
Sự kiện 3	<p>Flask server tiến hành phân loại dữ liệu theo “mẫu”. Có hai trường hợp:</p> <ul style="list-style-type: none"> <li>• TH 1: Server trả về danh sách chứa tên loại. Tiến hành sự kiện 4.</li> <li>• TH 2: Server trả về danh sách rỗng. Quay lại sự kiện 1 nhận dữ liệu là nội dung chính và tiến hành phân loại.</li> </ul>
Sự kiện 4	Lưu danh sách chứa tên loại.

*Bảng 14 Mô tả quá trình phân loại tin của Flask*

### 3.3 Quá trình đào tạo tập huấn luyện (tạo ra tập “mẫu”)



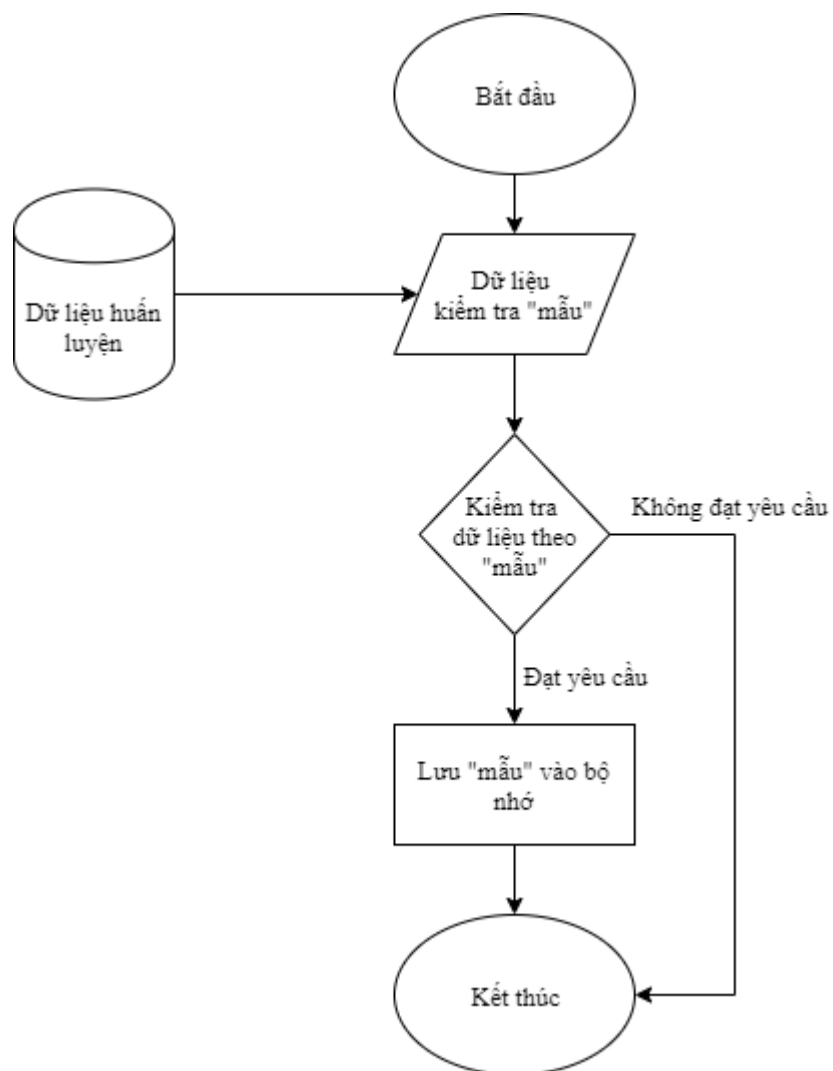
Hình 21 Quá trình tạo tập huấn luyện



<b>Luồng sự kiện</b>	
<b>Các sự kiện</b>	<b>Mô tả</b>
Sự kiện 1	Lấy một phần dữ liệu từ tập huấn luyện. Từ dữ liệu đã lấy, chia thành hai phần theo tỉ lệ 7:3, dùng phần chiếm tỉ lệ 7 để làm dữ liệu huấn luyện.
Sự kiện 2	Tải “mẫu” từ bộ nhớ làm “mẫu” huấn luyện, có hai trường hợp: <ul style="list-style-type: none"> <li>• Chưa có mẫu (huấn luyện lần đầu): Hệ thống sẽ tạo “mẫu” rỗng.</li> <li>• Đã có mẫu : Hệ thống sẽ tải mẫu đã có.</li> </ul>
Sự kiện 3	Tiến hành huấn luyện dữ liệu dựa vào “mẫu” đã tải lên.
Sự kiện 4	Cập nhật “mẫu” sau khi quá trình huấn luyện hoàn tất.
Sự kiện 5	Kiểm tra “mẫu” mới, có hai trường hợp: “Mẫu” đạt yêu cầu: Chuyển sang sự kiện 6. “Mẫu” không đạt yêu cầu: Quay lại sự kiện 1, lấy một phần dữ liệu (gồm dữ liệu đã huấn luyện và dữ liệu mới) huấn luyện tiếp. Quá trình sẽ lặp cho đến khi “mẫu” đạt yêu cầu.
Sự kiện 6	Lưu “mẫu” vào bộ nhớ và kết thúc quá trình huấn luyện.

*Bảng 15 Mô tả quá trình tạo tập “mẫu”*

❖ Quá trình kiểm tra mẫu.



Hình 22 Quá trình kiểm tra "mẫu"

Luồng sự kiện	
Các sự kiện	Mô tả
Sự kiện 1	Nhận dữ liệu kiểm tra “mẫu” là 3 phần dữ liệu còn lại của phần dữ liệu lớn trích từ tập huấn luyện.
Sự kiện 2	Tiến hành phân loại dữ liệu theo “mẫu” có được từ quá trình huấn luyện. Có hai trường hợp: <ul style="list-style-type: none"> <li>TH 1: Kết quả phân loại đúng như mong đợi. Chuyển sang sự kiện 3.</li> <li>TH 2: Kết quả phân loại không như mong đợi. Kết thúc quá trình kiểm tra.</li> </ul>
Sự kiện 3	Lưu “mẫu” vào bộ nhớ.

*Bảng 16 Mô tả quá trình kiểm tra tập “mẫu”*

❖ **Tại sao chia tập dữ liệu theo tỉ lệ 7:3 để làm tập huấn luyện và tập kiểm tra?**

Theo Andrew Ng, trong Coursera MOOC về Giới thiệu về Machine Learning, nguyên tắc chung là phân vùng dữ liệu thành tỷ lệ 3: 1: 1 (60:20:20) để đào tạo, xác nhận và kiểm tra tương ứng.

Khi một hệ thống học tập được đào tạo với một số mẫu dữ liệu, bạn có thể không biết chính xác đến mức nào tập huấn luyện có thể dự đoán các mẫu dữ liệu mới một cách chính xác. Khái niệm xác nhận chéo được thực hiện để điều chỉnh các tham số sử dụng cho quá trình đào tạo nhằm tối ưu hóa độ chính xác của tập huấn luyện và vô hiệu hóa hiệu quả của việc khớp quá mức trên dữ liệu đào tạo.

Trong trường hợp tập dữ liệu huấn luyện lớn sẽ không áp dụng xác nhận chéo (do quá trình xác nhận chéo rất tốn thời gian), vì vậy việc tách dữ liệu theo tỷ lệ 7: 3 (70:30) để đào tạo và kiểm tra độ chính xác được sử dụng phổ biến.

### **III. HỆ THỐNG HIỂN THỊ KẾT QUẢ**

#### **1. Tổng quan hệ thống**

Hệ thống hiển thị kết quả sử dụng các công cụ, thư viện, dịch vụ hỗ trợ quá trình đẩy kết quả lên trang hiển thị bao gồm: Ghost CMS, Nodejs server, Mobiledoc.

Ghost CMS là công cụ chủ yếu của hệ thống hiển thị kết quả, nó là một hệ thống quản trị nội dung được viết bằng ngôn ngữ lập trình Nodejs (Javascript). Ghost được sử dụng nhiều để tạo ra các blog, nhưng chúng ta có thể sử dụng, lập trình nó thành các website tin tức, website kinh doanh, ...

Nodejs server sử dụng trong hệ thống là một dịch vụ hỗ trợ chuyển đổi chuẩn nội dung tin tức từ chuẩn HTML sang chuẩn Mobiledoc.

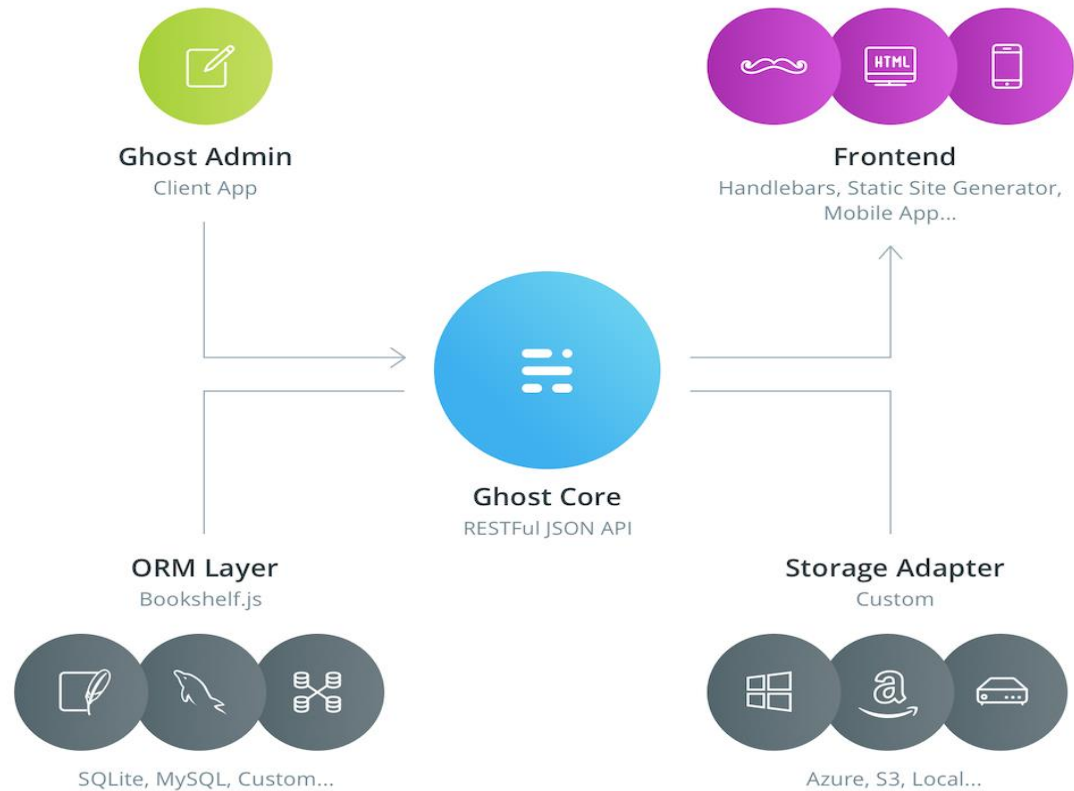
Mobiledoc là chuẩn dữ liệu trong hệ thống hiển thị tin tức vì Ghost CMS nhận dữ liệu đầu vào là chuẩn Mobiledoc.

Mobiledoc chủ yếu được sử dụng cho các nội dung liên quan đến tin tức như bài viết và bài đăng trên blog. Nó đơn giản có chủ ý và tổ chức nội dung của nó trong một mảng các “phần” được coi là các khối nội dung riêng lẻ.

Không có khái niệm về bố cục hoặc thiết kế được tích hợp trong Mobiledoc. Tùy thuộc vào trình kết xuất để tạo ra màn hình phù hợp với bối cảnh của nó. Trên thiết bị di động, điều này có nghĩa là mỗi phần có chiều rộng đầy đủ và chúng được hiển thị tuần tự. Trên màn hình lớn hơn, các phần có thể được hiển thị cạnh nhau. Mobiledoc không có quy tắc để hiển thị đầu ra.

## 2. Kiến trúc hệ thống

### 2.1 Kiến trúc Ghost CMS



Hình 23 Kiến trúc Ghost CMS

(nguồn: <https://ghost.org/docs/concepts/architecture/>)

Các thành phần trong kiến trúc Ghost CMS gồm:

Ghost Core: API JSONful RESTful - được thiết kế để tạo, quản lý và truy xuất nội dung xuất bản một cách dễ dàng.

Ghost Admin: Là trang quản trị của trang web sử dụng Ghost CMS.

Frontend: Là giao diện hiển thị của trang web sử dụng Ghost CMS.

ORM Layer: Là nơi lưu trữ dữ liệu.

Storage Adapter: Là phần cứng chứa cơ sở dữ liệu.

## 2.2 Cấu trúc chuẩn dữ liệu Mobiledoc

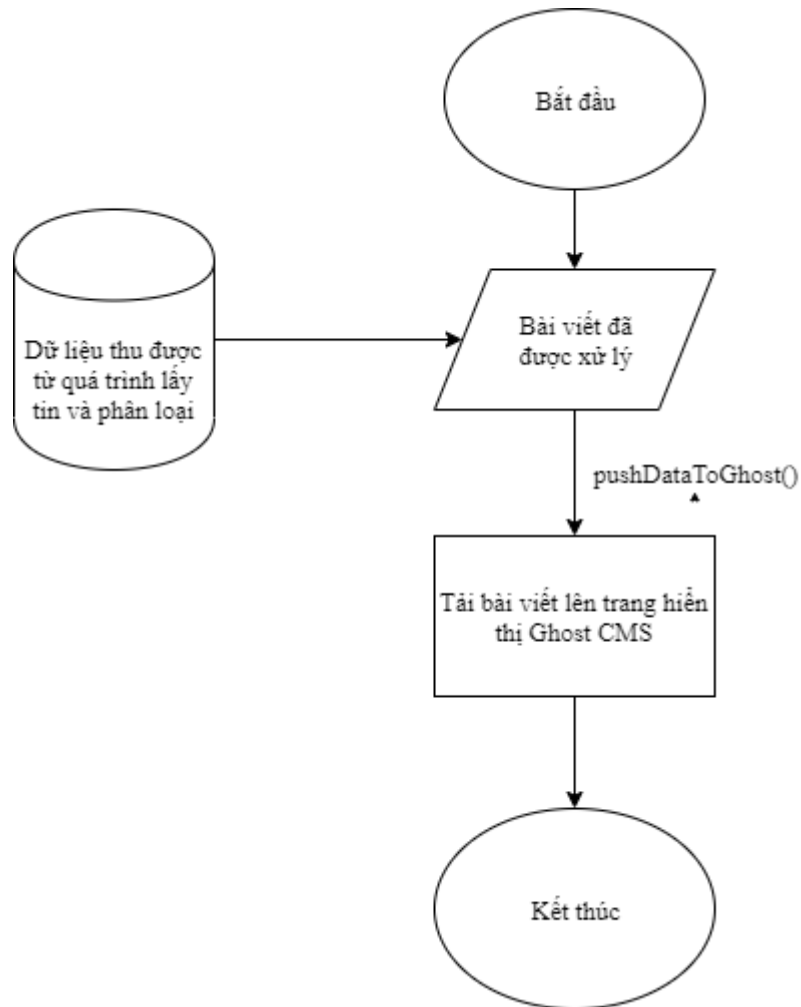
```
{
  version: "0.3.2",
  markups: [
    markup,
    markup
  ],
  atoms: [
    atom,
    atom
  ],
  cards: [
    card,
    card
  ],
  sections: [
    section,
    section,
    section
  ]
}
```

—— Versioning information  
—— Ordered list of markup types  
—— Ordered list of atom types  
—— Ordered list of card types  
—— Ordered list of sections.

Hình 24 Cấu trúc chuẩn dữ liệu Mobiledoc

(nguồn: <https://github.com/bustle/mobiledoc-kit/blob/master/MOBILEDOC.md>)

### 2.3 Sơ đồ quá trình hiển thị kết quả

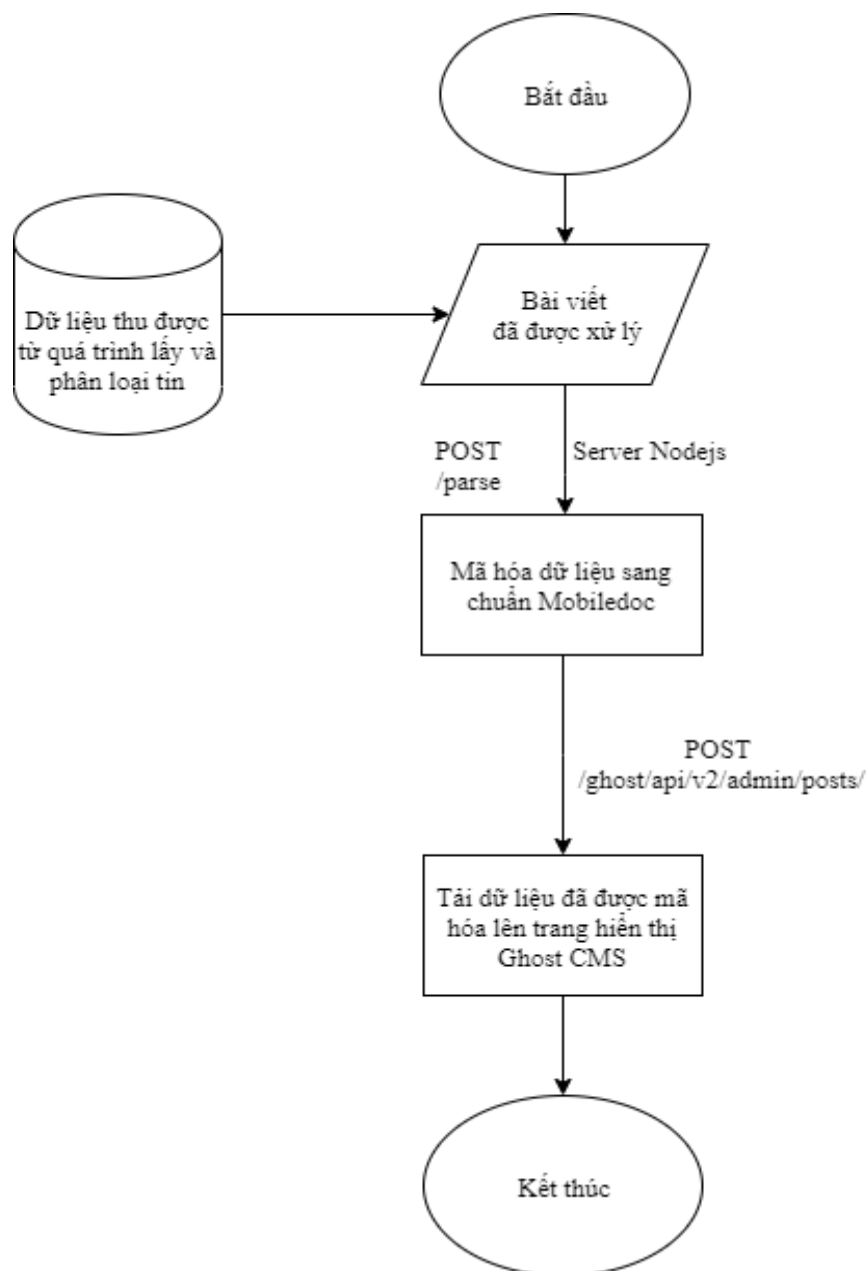


Hình 25 Sơ đồ quá trình tải dữ liệu lên Ghost CMS

Luồng sự kiện	
Các sự kiện	Mô tả
Sự kiện 1	Nhận dữ liệu thu được từ quá trình lấy và phân loại.
Sự kiện 2	Gọi hàm pushDataToGhost() để tải dữ liệu lên trang hiển thị.

Bảng 17 Mô tả quá trình hiển thị kết quả

### 3. Thiết kế hệ thống của quá trình hiển thị kết quả



Hình 26 Sơ đồ hoạt động của hệ thống hiển thị kết quả

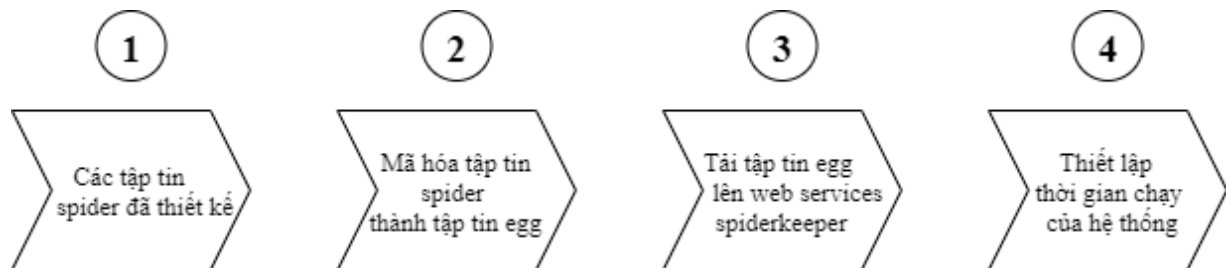


Luồng sự kiện	
Các sự kiện	Mô tả
Sự kiện 1	Gửi dữ liệu sang server Nodejs thông qua API POST /parse
Sự kiện 2	Server Nodejs nhận và tiến hành mã hóa dữ liệu sang chuẩn Mobiledoc.
Sự kiện 3	Dữ liệu đã được mã hóa sẽ được tải lên trang hiển thị Ghost CMS thông qua API của Ghost (POST /ghost/api/v2/admin/posts/).

Bảng 18 Mô tả hoạt động hệ thống hiển thị kết quả

#### IV. TỰ ĐỘNG HÓA HỆ THỐNG

##### ❖ Quy trình tự động hóa hệ thống



Hình 27 Các bước tự động hóa hệ thống

**Bước 1:** Thiết kế các tập tin spider cho hệ thống.

**Bước 2:** Mã hóa các tập tin spider thành tập tin egg.

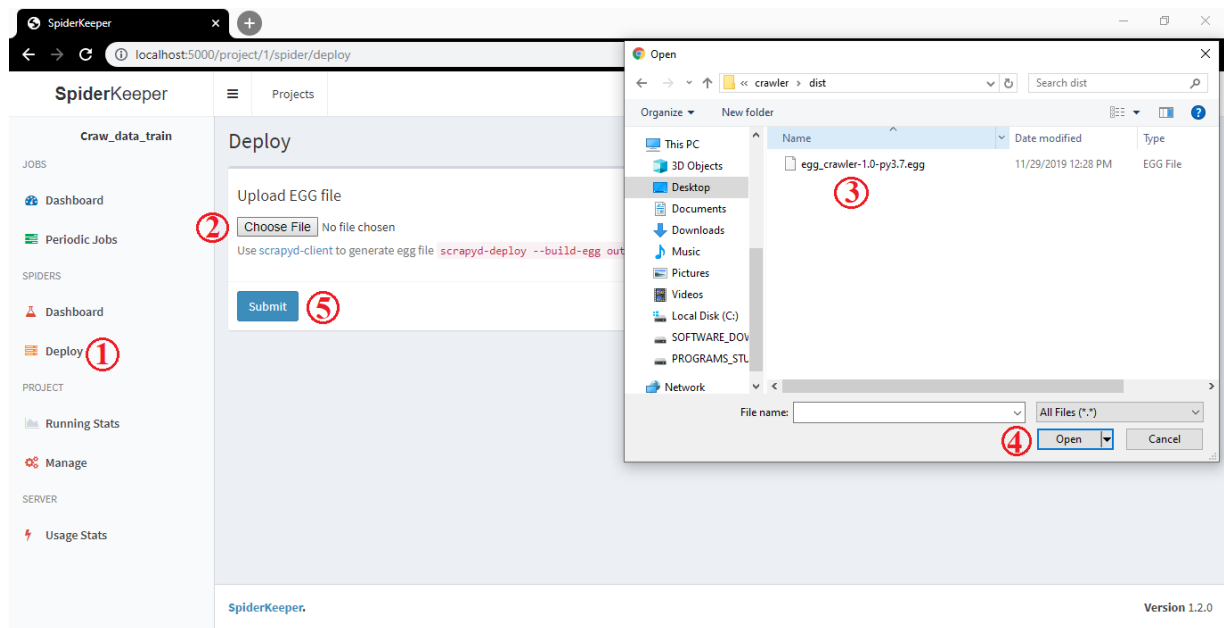
- ❖ Tạo tập tin setup.py để mã hóa các tập tin spider thành tập tin egg. Tập tin setup.py gồm nội dung:
  - Name: tên tập tin egg
  - Version: phiên bản tập tin egg
  - Packages: find\_packages()
  - Entry\_points
- ❖ Chạy tập tin setup.py để tạo tập tin egg từ các tập tin spider.

```
C:\Users\Toannt\Desktop\ScrapyProject\news-crawler-toannguyen-sangho\crawler>python setup.py bdist_egg
```

- ❖ Kiểm tra tập tin egg tạo thành công hay không bằng cách vào đường dẫn thư mục dự án / thư mục chứa tập tin spider / thư mục dist.

 egg_crawler-1.0-py3.7.egg	11/29/2019 12:28 PM	EGG File	21 KB
---	---------------------	----------	-------

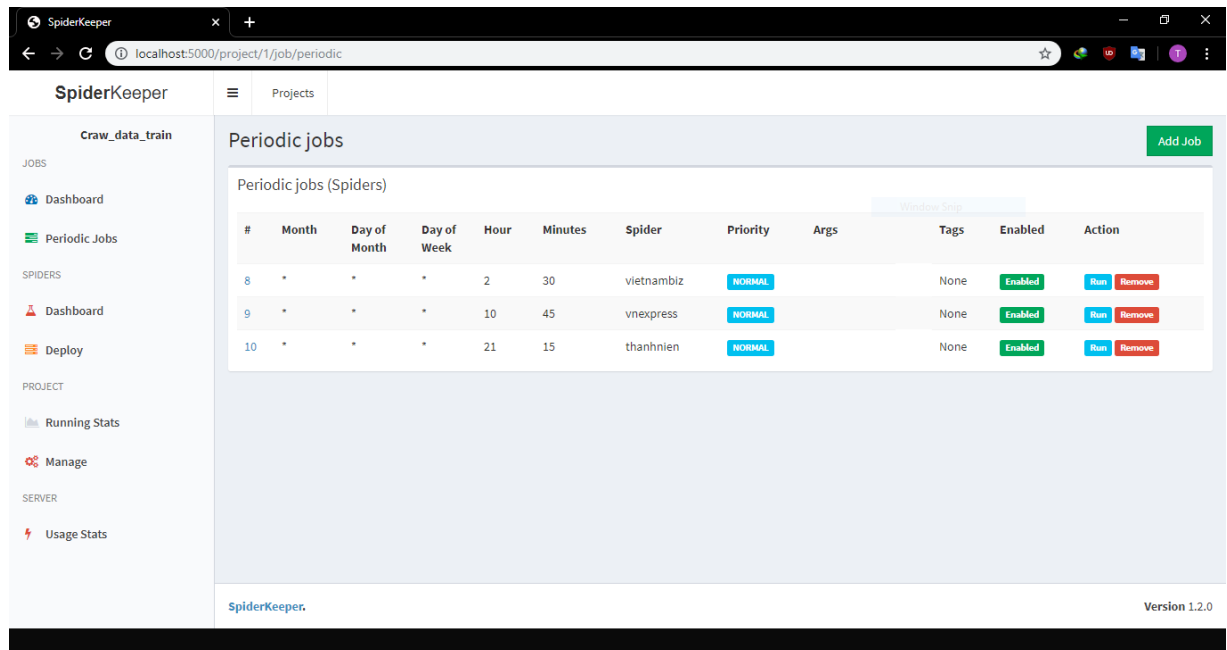
### Bước 3: Tải tập tin egg lên web services spiderkeeper.



Hình 28 Tải tập tin egg lên spiderkeeper

- Bước 1: Chọn “Deploy” để chuyển qua giao diện tải tập tin egg.
- Bước 2: Chọn “Choose File” để tải tập egg lên hệ thống.
- Bước 3: Chọn tập tin egg cần tải lên.
- Bước 4: Chọn “Open” để xác nhận tập tin egg cần tải lên.
- Bước 5: Chọn “Submit” để tiến hành tải lên spiderkeeper.

### Bước 4: Thiết lập thời gian chạy của hệ thống trên web services spiderkeeper.



Hình 29 Thiết lập thời gian chạy của hệ thống

## CHƯƠNG III KIỂM THỬ VÀ ĐÁNH GIÁ

### I. GIỚI THIỆU

#### 1. Mục tiêu kiểm thử và đánh giá

- Nhằm xác định các lỗi trong trường hợp có thể xảy ra của ứng dụng.
- Phát hiện lỗi phần mềm và kiểm tra hệ thống có hoạt động tốt đúng với yêu cầu đã nêu.
- Đảm bảo tính hoàn thiện của ứng dụng trước khi bàn giao sản phẩm cho khách hàng.
- Liệt kê các kết quả có được sau khi kiểm thử.
- Làm tài liệu cho giai đoạn bảo trì.

#### 2. Phạm vi kiểm thử

Do thời gian có hạn và các chức năng có phần tương đồng nên việc kiểm thử chỉ thực hiện trên một số chức năng so với toàn bộ các chức năng trên tài liệu đặc tả.

Quy trình kiểm thử được thực hiện qua các giai đoạn:

- Kiểm thử hệ thống: Kiểm tra thiết kế và hệ thống thoả mãn đặc tả.
- Kiểm thử chấp nhận: Kiểm tra chức năng hệ thống có thoả mãn các yêu cầu đặc tả hay không.
- Kiểm thử chức năng: Kiểm tra chức năng có xử lý đúng dữ liệu hay không.
- Kiểm thử cài đặt: Tìm và sửa các lỗi xảy ra khi kiểm thử.

### II. CHI TIẾT KẾ HOẠCH KIỂM THỬ

#### 1. Các chức năng sẽ được kiểm thử

Các chức năng sẽ được kiểm thử với hệ thống:

- Hoạt động của hệ thống lấy tin có ổn định.
- Tin tức được lấy về có chính xác.
- Tin tức phân loại có đạt yêu cầu đề ra.
- Tin tức hiển thị trên trang kết quả.

#### 2. Các chức năng không được kiểm thử

Những chức năng không được kiểm thử là những chức năng đơn giản, không có xử lý phức tạp hay các chức năng đó tương tự các chức năng đã được kiểm thử hoặc chức năng chưa hoàn thiện.

### **3. Cách tiếp cận**

Với mỗi tính năng chính hay nhóm tính năng sẽ được kiểm thử và được ghi nhận kết quả kiểm thử, đảm bảo rằng sẽ kiểm thử và không bỏ sót chức năng cần kiểm thử.

Tổ chức kiểm thử theo từng chức năng, mỗi chức năng được kiểm thử với các kịch bản kiểm thử và ghi nhận kết quả kiểm thử.

### **4. Tiêu chí kiểm thử thành công / thất bại**

- Tiêu chí kiểm thử thành công là kết quả thực hiện chức năng đúng với mong đợi, phù hợp với đặc tả yêu cầu.
- Tiêu chí kiểm thử thất bại là kết quả không như mong đợi, xuất hiện lỗi, không phù hợp với đặc tả yêu cầu.

### **5. Tiêu chí đình chỉ và yêu cầu bắt đầu làm lại**

Khi kiểm thử một chức năng có kết quả là một trang web rỗng, toàn code hoặc chờ đợi quá lâu thì phải dừng việc kiểm thử, chờ sửa lỗi và bắt đầu thực hiện lại chức năng đó và có thể phải kiểm thử một số chức năng liên quan.

### **6. Sản phẩm bàn giao của kiểm thử**

- Tài liệu kế hoạch kiểm thử.
- Các trường hợp kiểm thử.
- Các ghi chú kiểm thử.
- Báo cáo kiểm thử.
- Tự động hóa hệ thống.

### III. CÁC TRƯỜNG HỢP KIỂM THỬ

#### 1. Trường hợp kiểm thử 1: Hoạt động hệ thống lấy tin.

##### 1.1 Mục tiêu

Trường hợp kiểm thử này nhằm kiểm tra độ ổn định của hệ thống lấy tin.

##### 1.2 Kết nhập

Mã trường hợp	Nội dung	Ghi chú
TH_01	Chạy bất kì một “con” spider trong hệ thống lấy tin	
TH_02	Kiểm tra tập ghi chú sau mỗi lần chạy của các “con” spider	
TH_03	Các “con spider” có chạy đúng lịch biểu được thiết đặt	

*Bảng 19 Kết nhập trường hợp kiểm thử 1*

##### 1.3 Kết xuất

Mã trường hợp	Mục tiêu mong đợi	Mục tiêu thực tế	Ghi chú
TH_01	“Con” spider hoạt động	“Con” spider hoạt động	Thành công
TH_02	Tập ghi chú chứa chi tiết nội dung chạy của spider	Tập ghi chú có chứa nội dung spider vừa chạy	Thành công
TH_03	Thời gian chạy của spider đúng với lịch biểu	Thời gian chạy của spider đúng với lịch biểu	Thành công

*Bảng 20 Kết xuất trường hợp kiểm thử 1*

## 2. Trường hợp kiểm thử 2: Độ chính xác của nội dung tin tức.

### 2.1 Mục tiêu

Trường hợp kiểm thử này nhằm kiểm định độ chính xác về mặt nội dung tin tức mà các “con” spider lấy về.

### 2.2 Kết nhập

Mã trường hợp	Nội dung	Ghi chú
TH_01	So sánh bố cục trang hiển thị một bài tin với bài tin nguồn.	
TH_02	Kiểm tra nội dung đặc biệt trong bài như chữ in nghiêng, in đậm, hình ảnh	

*Bảng 21 Kết nhập trường hợp kiểm thử 2*

### 2.3 Kết xuất

Mã trường hợp	Mục tiêu mong đợi	Mục tiêu thực tế	Ghi chú
TH_01	Bố cục hiển thị giữa trang nguồn và trang kết quả như nhau	Bố cục hiển thị giữa trang nguồn và trang kết quả giống nhau	Thành công
TH_02	Các nội dung đặc biệt như chữ in đậm, in nghiêng, hình ảnh hiển thị đầy đủ.	Các nội dung đặc biệt như chữ in đậm, in nghiêng, hình ảnh hiển thị đầy đủ.	Thành công

*Bảng 22 Kết xuất trường hợp kiểm thử 2*

### 3. Trường hợp kiểm thử 3: Khả năng phân loại của hệ thống

#### 3.1 Mục tiêu

Trường hợp kiểm thử này nhằm mục đích xác định hệ thống phân loại tự động có hoạt động không và hoạt động có được chính xác như mục tiêu đề ra hay không.

#### 3.2 Kết nhập

Mã trường hợp	Nội dung	Ghi chú
TH_01	Kiểm tra bài tin sau khi phân loại có tên loại (tag) mới không.	
TH_02	Độ chính xác của tên loại bài tin có như mong đợi.	
TH_03	Có bài tin nào không có tên loại hay không.	

*Bảng 23 Kết nhập trường hợp kiểm thử 3*

#### 3.3 Kết xuất

Mã trường hợp	Mục tiêu mong đợi	Mục tiêu thực tế	Ghi chú
TH_01	Bài tin sau khi phân loại có tên loại mới.	Bài tin sau khi phân loại có tên loại mới.	Thành công
TH_02	Tên loại của bài tin như mong đợi.	Tên loại của bài tin như mong đợi.	Thành công
TH_03	Tất cả các bài tin đều có tên loại mới.	Tất cả các bài tin đều có tên loại mới.	Thành công

*Bảng 24 Kết xuất trường hợp kiểm thử 3*



#### 4. Trường hợp kiểm thử 4: Trang hiển thị kết quả

##### 4.1 Mục tiêu

Trường hợp kiểm thử này nhằm xác định trang kết quả có hiển thị nội dung đúng như yêu cầu đề ra hay không.

##### 4.2 Kết nhập

Mã trường hợp	Nội dung	Ghi chú
TH_01	Trang kết quả hiển thị đầy đủ tin lấy về không.	
TH_02	Nội dung hiển thị có đúng với nội dung tin nguồn không.	
TH_03	Thứ tự hiển thị của các bài tin có đúng với thứ tự tin lấy về không.	

*Bảng 25 Kết nhập trường hợp kiểm thử 4*

##### 4.3 Kết xuất

Mã trường hợp	Mục tiêu mong đợi	Mục tiêu thực tế	Ghi chú
TH_01	Trang kết quả hiển thị đầy đủ tin tức.	Trang kết quả hiển thị đầy đủ tin tức.	Thành công
TH_02	Nội dung mỗi bài tin đúng với nội dung nguồn.	Nội dung mỗi bài tin đúng với nội dung nguồn.	Thành công
TH_03	Thứ tự hiển thị các bài tin đúng với thứ tự lấy tin về.	Thứ tự hiển thị các bài tin đúng với thứ tự lấy tin về.	Thành công

*Bảng 26 Kết xuất trường hợp kiểm thử 4*

## 5. Trường hợp kiểm thử 5 : Tự động hóa hệ thống

### 5.1 Mục tiêu

Trường hợp kiểm thử này nhằm xác định hệ thống có hoạt động tự động chính xác theo lịch biểu được thiết đặt.

### 5.2 Kết nhập

Mã trường hợp	Nội dung	Ghi chú
TH_01	Hệ thống có tự động hoạt động chính xác theo lịch biểu đã được thiết đặt	

*Bảng 27 Kết nhập trường hợp kiểm thử 5*

### 5.3 Kết xuất

Mã trường hợp	Mục tiêu mong đợi	Mục tiêu thực tế	Ghi chú
TH_01	Hệ thống tự động hoạt động chính xác theo lịch biểu được thiết đặt	Hệ thống tự động hoạt động chính xác theo lịch biểu được thiết đặt	Thành công

*Bảng 28 Kết xuất trường hợp kiểm thử 5*

#### IV. KẾT QUẢ KIỂM THỬ

STT	Tên chức năng	Số trường hợp kiểm thử	Số lần thành công	Số lần thất bại	Ghi chú
1	Hoạt động của hệ thống lấy tin	3	3	0	
2	Độ chính xác của nội dung tin tức	2	2	0	
3	Khả năng phân loại của hệ thống	3	3	0	
4	Trang hiển thị kết quả	3	3	0	
5	Tự động hóa hệ thống	1	1	0	

*Bảng 29 Kết quả kiểm thử*

## PHẦN KẾT LUẬN

### I. KẾT QUẢ ĐẠT ĐƯỢC

#### 1. Về lý thuyết

Về lý thuyết sau thời gian nghiên cứu, tự tìm hiểu công nghệ, kiến thức chuyên môn để thực hiện đề tài đã giúp cho người thực hiện đề tài bổ sung vốn kiến thức lập trình cho bản thân, có cái nhìn tổng quan về quy trình phát triển phần mềm là như thế nào, từ các khâu phân tích, thiết kế đến lập trình và kiểm thử một ứng dụng.

Bên cạnh đó, còn học hỏi được rất nhiều những tiện ích khi sử dụng các công cụ Visual Studio Code, PowerDesigner, Docker ... vào việc phát triển ứng dụng.

Trọng tâm là xây dựng hệ thống lấy tin tức và phân loại tin tự động, trong đề tài sử dụng Python, Docker, quan trọng là sử dụng máy học áp dụng vào phân loại tin tức để phát triển ứng dụng, qua đó hiểu biết thêm cơ bản về công nghệ đó.

Hiểu rõ được các quy trình cơ bản về các hoạt động liên quan đến mảng báo chí và tin tức trực tuyến trên Internet, giúp nâng cao tầm hiểu biết của bản thân.

#### 2. Về chương trình

Xây dựng thành công một hệ thống tự động về lấy tin, phân loại tin và hiển thị tin tức lên website, đáp ứng đầy đủ các chức năng đề ra ban đầu của đề tài.

Hệ thống đã đáp ứng tốt các yêu tố yêu cầu sau:

- Hệ thống vận hành tự động.
- Thời gian lấy tin được cài đặt lịch biểu và hệ thống hoạt động theo lịch biểu đó.
- Chức năng phân loại tin hoạt động tốt và độ chính xác thỏa mãn yêu cầu.
- Trang hiển thị kết quả hiển thị đầy đủ yêu cầu đề ra.

Hệ thống hoạt động tốt đáp ứng các yêu cầu cần của một website tin tức có ứng công nghệ vào thực tế.

#### 3. Khả năng ứng dụng thực tiễn

Hệ thống có thể ứng dụng vào thực tế, tạo ra một trang tin tức vận hành tự động, có khả năng thu thập tin tức mới nhất một cách nhanh chóng với nguồn tin được tổng hợp từ nhiều trang báo điện tử khác nhau.

## II. HẠN CHẾ

- Công nghệ mới, nguồn tài liệu còn hạn hẹp.
- Mất nhiều thời gian để học và làm quen với ngôn ngữ, Framework mới.
- Một số giao diện còn chưa thân thiện.
- Chưa thể nhận phản hồi từ người dùng.
- Khả năng phân loại còn hạn chế nên gặp nhiều khó khăn.
- Một số tính năng còn hạn chế, chưa đạt hiệu quả cao.

## III. HƯỚNG PHÁT TRIỂN

Do điều kiện thời gian còn hạn chế, nên vấn đề nghiên cứu và thực hiện đề tài “*Xây dựng hệ thống lấy tin và phân loại tin tự động*” trong khuôn khổ của luận văn mới chỉ dừng lại ở những nghiên cứu cơ bản.

Hướng phát triển của đề tài:

- Tăng cường hiệu năng, bảo mật cho hệ thống khi ứng dụng vào thực tế.
- Nâng cấp và cập nhật thêm các chức năng phù hợp với nhu cầu của người sử dụng.
- Phát triển chạy trên đa nền tảng (ứng dụng dành riêng cho nền tảng di động).
- Tối ưu hóa cơ sở dữ liệu và áp dụng giải thuật để nâng cấp hệ thống xử lý dữ liệu tốt hơn.

## TÀI LIỆU THAM KHẢO

[1] Ngôn ngữ lập trình Python:

Sách Coding project in Python (Tác giả Ben Morgan, senior editors, NXB New York - Penguin Random House, năm 2017 )

<https://www.python.org/>

<https://www.w3schools.com/python/>

[2] Nodejs express

Sách Sổ tay HTML và Javascript (Tác giả Nguyễn Trường Sinh, NXB Hà Nội – Lao động xã hội, năm 2006)

Sách Web Development with Node and Express (Tác giả Ethan Brown, NXB O'reilly, năm 2014)

<https://expressjs.com/>

[https://www.tutorialspoint.com/nodejs/nodejs\\_express\\_framework.htm](https://www.tutorialspoint.com/nodejs/nodejs_express_framework.htm)

[3] Docker

<https://www.docker.com/>

<https://docs.docker.com/>

<https://hub.docker.com/>

[4] Scrapy và trình thu thập web

<https://scrapy.org/>

<https://docs.scrapy.org/en/latest/>

[https://en.wikipedia.org/wiki/Web\\_crawler](https://en.wikipedia.org/wiki/Web_crawler)

[5] MongoDB

<https://www.mongodb.com/>

<https://viblo.asia/p/mongodb-la-gi-co-so-du-lieu-phi-quan-he-bJzKmgoPl9N>

[6] RSS

[https://vi.wikipedia.org/wiki/RSS\\_\(%C4%91%E1%BB%8Bnh\\_d%E1%BA%A1ng\\_t%E1%BA%ADp\\_tin\)](https://vi.wikipedia.org/wiki/RSS_(%C4%91%E1%BB%8Bnh_d%E1%BA%A1ng_t%E1%BA%ADp_tin))

[7] Ghost CMS

<https://ghost.org/>

<https://vinasupport.com/ghost-cms-nodejs-open-source-platform-cho-website-blog/>

[https://en.wikipedia.org/wiki/Ghost\\_\(blogging\\_platform\)](https://en.wikipedia.org/wiki/Ghost_(blogging_platform))

[8] Nguyên lý máy học

Giáo trình Nguyên lý máy học (Tác giả TS. Đỗ Thanh Nghị - TS. Phạm Nguyên Khang, NXB Đại học Cần Thơ).

[9] spaCy

<https://spacy.io/>

[10] Mobiledoc

<http://bustle.github.io/mobiledoc-kit/demo/>

<https://github.com/bustle/mobiledoc-kit/blob/master/MOBILEDOC.md>

[11] Flask

<https://www.fullstackpython.com/flask.html>

<https://www.tutorialspoint.com/flask/index.htm>

[12] Spiderkeeper

<https://github.com/DormyMo/SpiderKeeper>

[13] Các nguồn tham khảo khác

- Các báo cáo luận văn của sinh viên khóa trước tại thư viện Khoa Công Nghệ Thông Tin và Truyền Thông trường Đại học Cần Thơ.
- <https://stackoverflow.com/>

## PHỤ LỤC

### I. CÁC CÔNG CỤ CHÍNH TRONG HỆ THỐNG

#### 1. Ngôn ngữ lập trình Python

Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Đặc điểm chính và nổi bật nhất ở Python là **dễ học và dễ đọc**.

Python được thiết kế để trở thành một ngôn ngữ dễ học, mã nguồn dễ đọc, bố cục trực quan, dễ hiểu, thể hiện qua các điểm sau:

##### Về từ khóa:

- Python tăng cường sử dụng từ khóa tiếng Anh, hạn chế các ký hiệu và cấu trúc cú pháp so với các ngôn ngữ khác.
- Python là một ngôn ngữ phân biệt kiểu chữ HOA, chữ thường.
- Như C/C++, các từ khóa của Python đều ở dạng chữ thường.

##### Về khối lệnh:

- Trong các ngôn ngữ khác, khối lệnh thường được đánh dấu bằng cặp ký hiệu hoặc từ khóa. Ví dụ, trong C/C++, cặp ngoặc nhọn { } được dùng để bao bọc một khối lệnh. Python, trái lại, có một cách rất đặc biệt để tạo khối lệnh, đó là thụt các câu lệnh trong khối vào sâu hơn (về bên phải) so với các câu lệnh của khối lệnh cha chứa nó.

##### Về các bản hiện thực:

- Python được viết từ những ngôn ngữ khác, tạo ra những bản hiện thực khác nhau. Bản hiện thực Python chính, còn gọi là CPython, được viết bằng C, và được phân phối kèm một thư viện chuẩn lớn được viết hỗn hợp bằng C và Python. CPython có thể chạy trên nhiều nền và khả năng chuyển trên nhiều nền khác. Dưới đây là các nền trên đó, CPython có thể chạy.
- Các hệ điều hành họ Unix: AIX, Darwin, FreeBSD, Mac OS X, NetBSD, Linux, OpenBSD, Solaris,...
- Các hệ điều hành dành cho máy tính để bàn: Amiga, AROS, BeOS, Mac OS 9, Microsoft Windows, OS/2, RISC OS.



- Các hệ thống nhúng và các hệ đặc biệt: GP2X, Máy ảo Java, Nokia 770 Internet Tablet, Palm OS, PlayStation 2, PlayStation Portable, Psion, QNX, Sharp Zaurus, Symbian OS, Windows CE/Pocket PC, Xbox/XBMC, VxWorks.
- Các hệ máy tính lớn và các hệ khác: AS/400, OS/390, Plan 9 from Bell Labs, VMS, z/OS.
- Ngoài CPython, còn có hai hiện thực Python khác: Jython cho môi trường Java và IronPython cho môi trường .NET và Mono.

#### **Về khả năng mở rộng:**

- Python có thể được mở rộng: nếu ta biết sử dụng C, ta có thể dễ dàng viết và tích hợp vào Python nhiều hàm tùy theo nhu cầu. Các hàm này sẽ trở thành hàm xây dựng sẵn (built-in) của Python. Ta cũng có thể mở rộng chức năng của trình thông dịch, hoặc liên kết các chương trình Python với các thư viện chỉ ở dạng nhị phân (như các thư viện đồ họa do nhà sản xuất thiết bị cung cấp). Hơn thế nữa, ta cũng có thể liên kết trình thông dịch của Python với các ứng dụng viết từ C và sử dụng nó như là một mở rộng hoặc một ngôn ngữ dòng lệnh phụ trợ cho ứng dụng đó.

#### **Về lệnh cấu trúc và điều khiển:**

- Mỗi câu lệnh trong Python nằm trên một dòng mã nguồn. Ta không cần phải kết thúc câu lệnh bằng bất kỳ ký tự gì. Cũng như các ngôn ngữ khác, Python cũng có các cấu trúc điều khiển.

Chúng bao gồm:

- Cấu trúc rẽ nhánh: cấu trúc if (có thể sử dụng thêm elif hoặc else), dùng để thực thi có điều kiện một khối mã cụ thể.
- Cấu trúc lặp, bao gồm:
- Lệnh while: chạy một khối mã cụ thể cho đến khi điều kiện lặp có giá trị false.
- Vòng lặp for: lặp qua từng phần tử của một dãy, mỗi phần tử sẽ được đưa vào biến cục bộ để sử dụng với khối mã trong vòng lặp.
- Python cũng có từ khóa class dùng để khai báo lớp (sử dụng trong lập trình hướng đối tượng) và lệnh def dùng để định nghĩa hàm.

#### **Về hệ thống kiểu dữ liệu:**

- Python sử dụng hệ thống kiểu duck typing, còn gọi là latent typing (tự động xác định kiểu). Có nghĩa là, Python không kiểm tra các ràng buộc

về kiểu dữ liệu tại thời điểm dịch, mà là tại thời điểm thực thi. Khi thực thi, nếu một thao tác trên một đối tượng bị thất bại, thì có nghĩa là đối tượng đó không sử dụng một kiểu thích hợp.

Python cũng là một ngôn ngữ định kiểu mạnh. Nó cấm mọi thao tác không hợp lệ, ví dụ cộng một con số vào chuỗi ký tự.

## 2. Docker

Docker là một nền tảng mở dành cho các lập trình viên, quản trị hệ thống dùng để xây dựng, vận chuyển và chạy các ứng dụng phân tán.

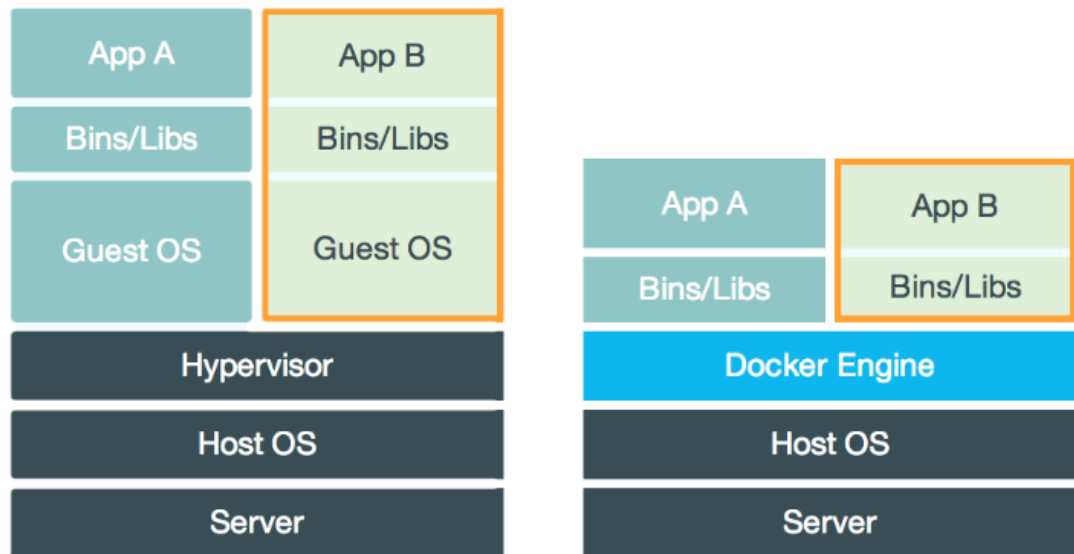
Docker có 4 thành phần cơ bản:

- Image : Là file ảnh, file nền của một hệ điều hành, một nền tảng, một ngôn ngữ hoặc một service
- Container : Là một máy ảo, được cấu thành từ 1 image. Các container này sẽ dùng chung tài nguyên của hệ thống (RAM, Disk, Network...), chính nhờ vậy, những container của bạn sẽ rất nhẹ, việc khởi động, kết nối, tương tác sẽ rất nhanh gọn. Nếu ánh xạ sang hướng đối tượng, thì image chính là class, còn container chính là instance-1 thể hiện của class đó. Từ 1 class ta có thể tạo ra nhiều instance, tương tự, từ 1 image ta cũng có thể tạo ra được nhiều container hoàn toàn giống nhau.
- Docker Engine : quản lý việc bạn tạo image, chạy container, dùng image có sẵn hay tải image chưa có về, kết nối vào container, thêm, sửa, xóa image và container, ...
- Docker Hub : Là 1 trang chia sẻ các image như github.

Một số khái niệm trong Docker:

- Docker images : là một “read-only template”. Chẳng hạn, một image chứa hệ điều hành Ubuntu đã cài đặt sẵn Apache và ứng dụng web
- Docker registries : Là kho chứa images. Người dùng có thể tạo ra các images của mình và tải lên đây hoặc tải về các images được chia sẻ
- Docker container : hoạt động giống như một thư mục (directory), chứa tất cả những thứ cần thiết để một ứng dụng có thể chạy được. Mỗi một docker container được tạo ra từ một docker image. Các thao tác với một container : chạy, bật, dừng, di chuyển, và xóa
- Dockerfile : là một file chứa tập hợp các lệnh để Docker có thể đọc và thực hiện để đóng gói một image theo yêu cầu người dùng
- Orchestration : là các công cụ, dịch vụ dùng để điều phối và quản lý nhiều containers sao cho chúng làm việc hiệu quả nhất

### So sánh giữa Docker và Virtual Machine



Điểm khác biệt chính là các containers sử dụng chung kernel với Host OS nên các thao tác bật, tắt rất nhẹ nhàng, nhanh chóng.

Ưu điểm : nhanh, nhẹ, có thể chia sẻ dễ dàng qua DockerHub.

Nhược điểm : mới, cập nhật thay đổi thường xuyên.

### 3. Scrapy và Web Crawler

Web Crawler là chương trình được thiết kế với mục đích có thể duyệt website trên mạng World Wide Web một cách có hệ thống, giúp thu thập thông tin của những trang web đó về cho công cụ tìm kiếm. Các tên gọi khác của crawler là robot, bot, spider, worm, ant, tuy nhiên gần đây tên gọi crawler hay spider vẫn là thông dụng nhất.

#### Spider nghĩa là gì?

Spider là cách gọi hình tượng hóa của Web Crawler, cái tên này được gọi dựa trên nguyên lý hoạt động và lưu thông tin của Web Crawler rất giống với những hoạt động của một con nhện. Bắt đầu từ một website bất kỳ, Spider sẽ len lỏi vào từng ngóc ngách ở trong trang đó và lần lượt truy cập vào từng liên kết có trên trang.

Sau đó nó sẽ đánh dấu các liên kết đã truy cập trước đó và nối các trang có liên kết với trang gốc giống như việc tạo một sợi tơ liên kết 2 trang lại với nhau. Chỉ đơn giản từ một website ban đầu, Spider có thể nối thêm rất nhiều website lại để tạo nên một mạng lưới chằng chịt như một mạng nhện đích thực.

### **Cách gọi Crawler là gì?**

Crawler là cách gọi theo chức năng của Web Crawler, tên gọi này có thể mô tả các hành động truy cập và thu thập dữ liệu của Web Crawler trên một website giống như một người hoặc một con bọ đang bò trườn trên trang đó.

### **Cơ chế hoạt động của Web Crawler là gì?**

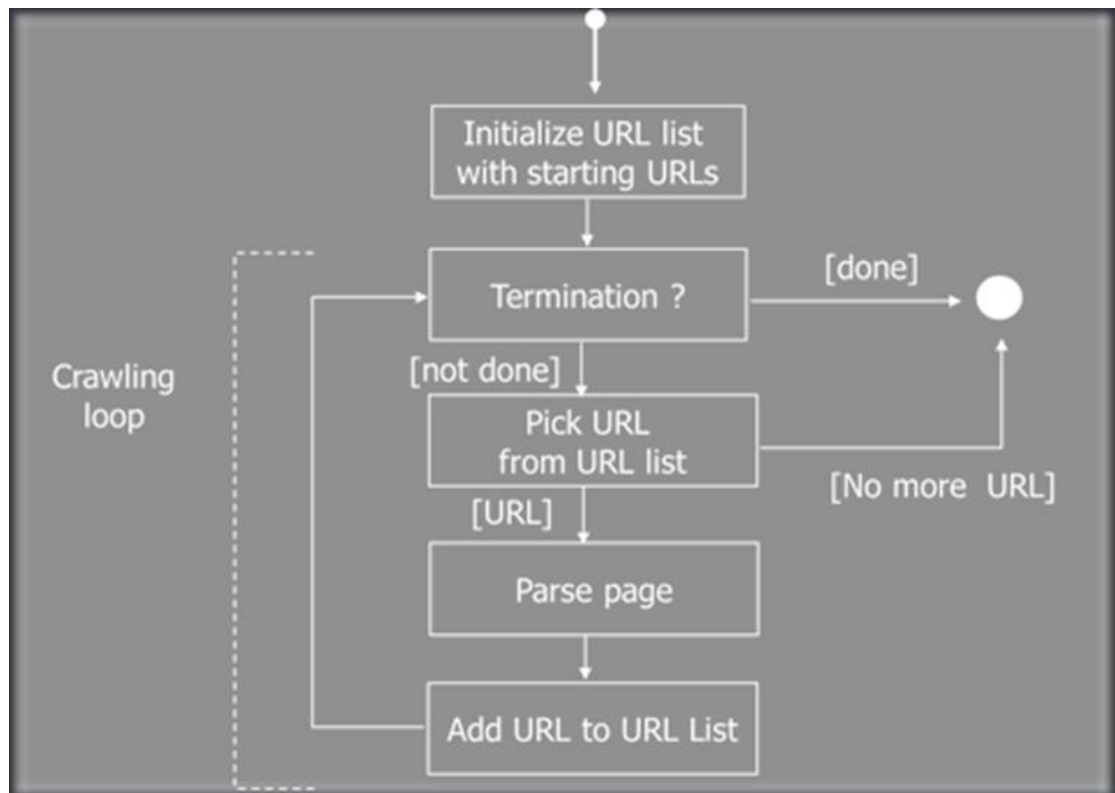
Web Crawler sở hữu tính năng khám phá và tìm hiểu thông tin trên các trang web công khai hiện nay trên mạng WWW. Các công cụ thu thập thông tin hữu ích này sẽ lần lượt theo dõi các trang web và dò theo từng liên kết trên các trang đó.

Nó cũng giống như việc chúng ta duyệt từng nội dung có trên trang. Web Crawler thu thập dữ liệu trên các trang bằng việc lần lượt đi từ liên kết này tới liên kết khác và đưa các dữ liệu đó về cho máy chủ Search Engine.

Để có thể crawl được các dữ liệu trên trang web, chúng ta cần quan tâm đến yếu tố đầu tiên đó là trang web bạn muốn crawl có bị chặn request hay không. Sau đó là vấn đề trang web bạn muốn crawl có cấu trúc có ổn định hay không?

Một trang web có cấu trúc ổn định sẽ dễ dàng để lấy data hơn là một trang web cấu trúc mỗi trang một định dạng khác nhau. Bởi lẽ khi chúng ta crawl sẽ chủ yếu dựa trên các element để lấy được data.

### Sơ đồ hoạt động của quá trình crawler



## 4. RSS

Khi nội dung internet trở nên phức tạp hơn, các tệp RSS cũng nhanh chóng chấp nhận hình ảnh, video..vv...nhưng vẫn ở định dạng rút gọn để tải và tương thích dễ dàng hơn trên tất cả các trình đọc nguồn cấp dữ liệu. Độc giả thường tự động cập nhật, để nó cung cấp nội dung mới nhất ngay cho thiết bị. Về cơ bản, phương pháp này cho phép người dùng internet tạo nguồn cấp dữ liệu trực tuyến riêng, chứa đầy các cập nhật tùy chỉnh từ các trang web muốn truy cập thường xuyên.

### Tại sao sử dụng RSS để crawl tin ?

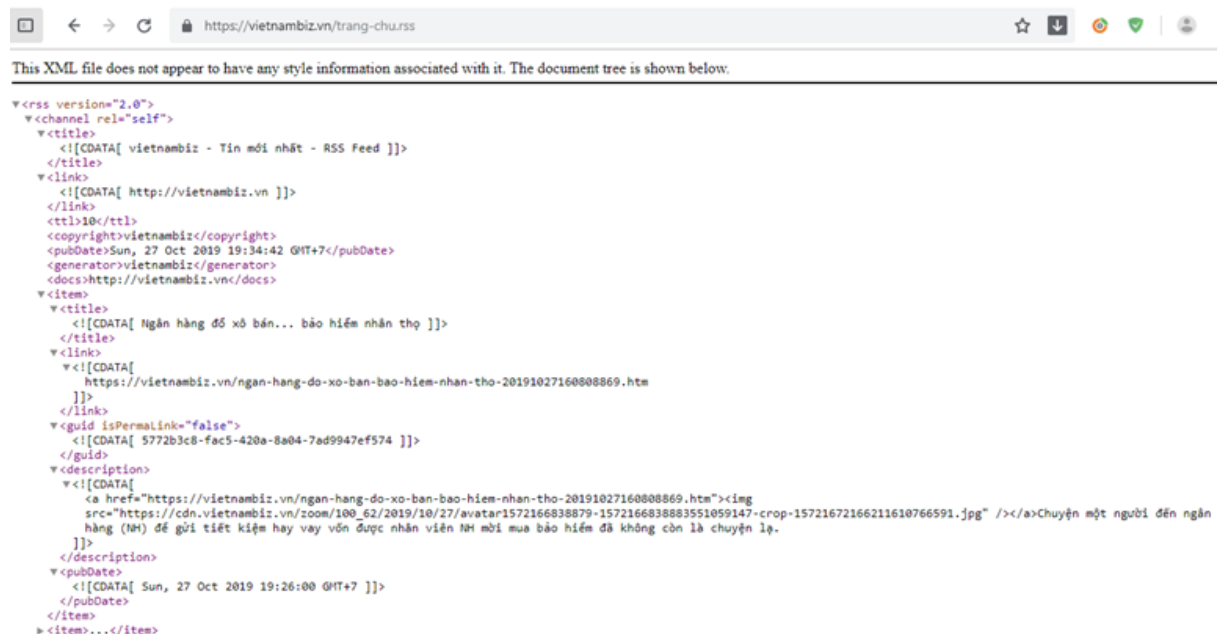
Hiện nay hầu hết những trang thông tin lớn như dantri, thanhnien, vnexpress, v.v... đều hỗ trợ truy cập bằng RSS. Việc thu thập những bài tin mới đăng từ các trang báo này cực kỳ dễ dàng vì hầu như mọi trang đều có những trang RSS công khai truy cập như trangchu.rss, tinmoinhat.rss. Chúng ta có thể dò ra được những bài báo mới đăng dễ dàng nhờ việc truy cập những trang RSS này.

Với sự phát triển của công nghệ hiện nay, đại đa số các trang báo lớn thường dùng chung một định dạng cho tất cả những bài báo của mình, trừ một vài trường hợp ngoại lệ. Và từ RSS chúng ta có thể trích xuất được tất cả những liên kết bài viết mới

cập nhật của trang báo. Từ những liên kết ấy và định dạng cố định của trang tin, chúng ta có thể lấy tin tức về một cách cực kỳ đơn giản.

Một trang RSS cơ bản bao gồm:

- Phiên bản của RSS (trong ảnh là version 2.0)
- Thẻ <title> : Tiêu đề của trang RSS
- Thẻ <link> : Đường dẫn đến trang chủ của trang tin RSS
- Thẻ <item> : Chứa các liên kết URL đến trang tin tức



## 5. Postman



Postman là một công cụ cho phép chúng ta làm việc với API, nhất là REST. Với Postman, ta có thể gọi Rest API mà không cần viết dòng code nào. Làm chủ Postman, bạn sẽ thấy việc gọi các Rest API (như Facebook, Google, Youtube) chả có gì phức tạp cả.

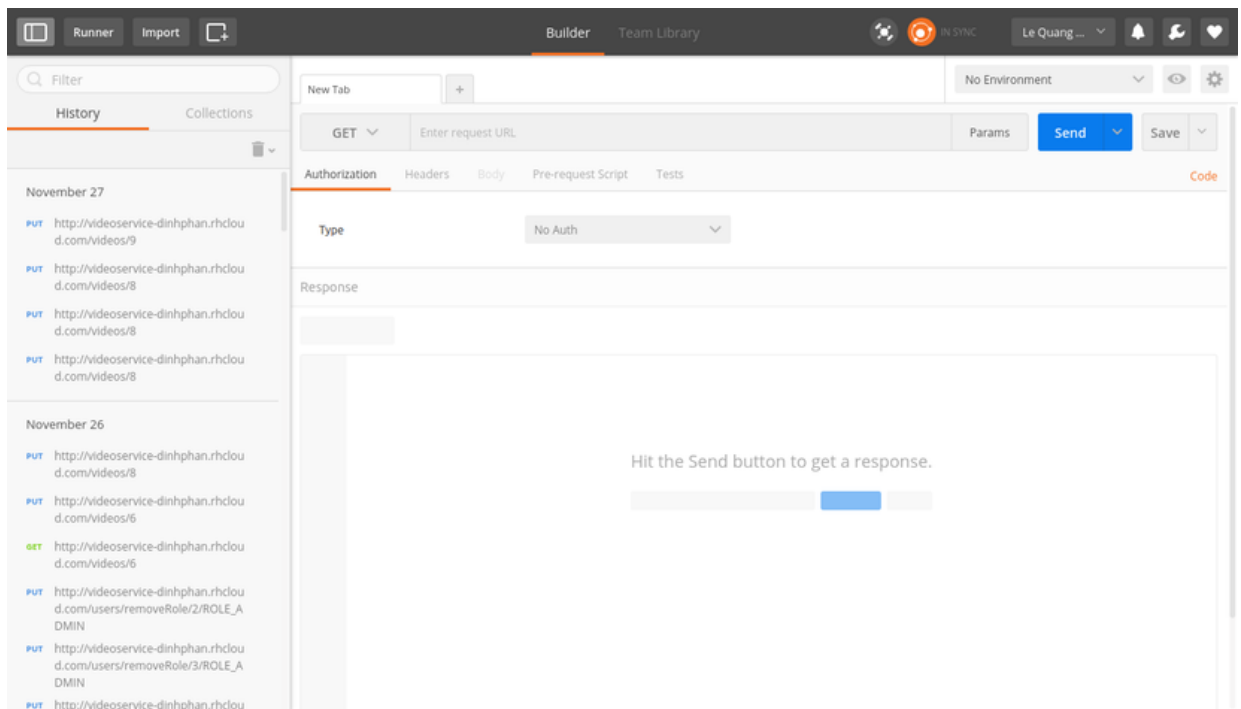
Postman hỗ trợ tất cả các phương thức HTTP (GET, POST, PUT, PATCH, DELETE, ...).

Postman cho phép lưu lại lịch sử các lần request, rất tiện cho việc sử dụng lại khi cần.

### Các chức năng cơ bản của Postman:

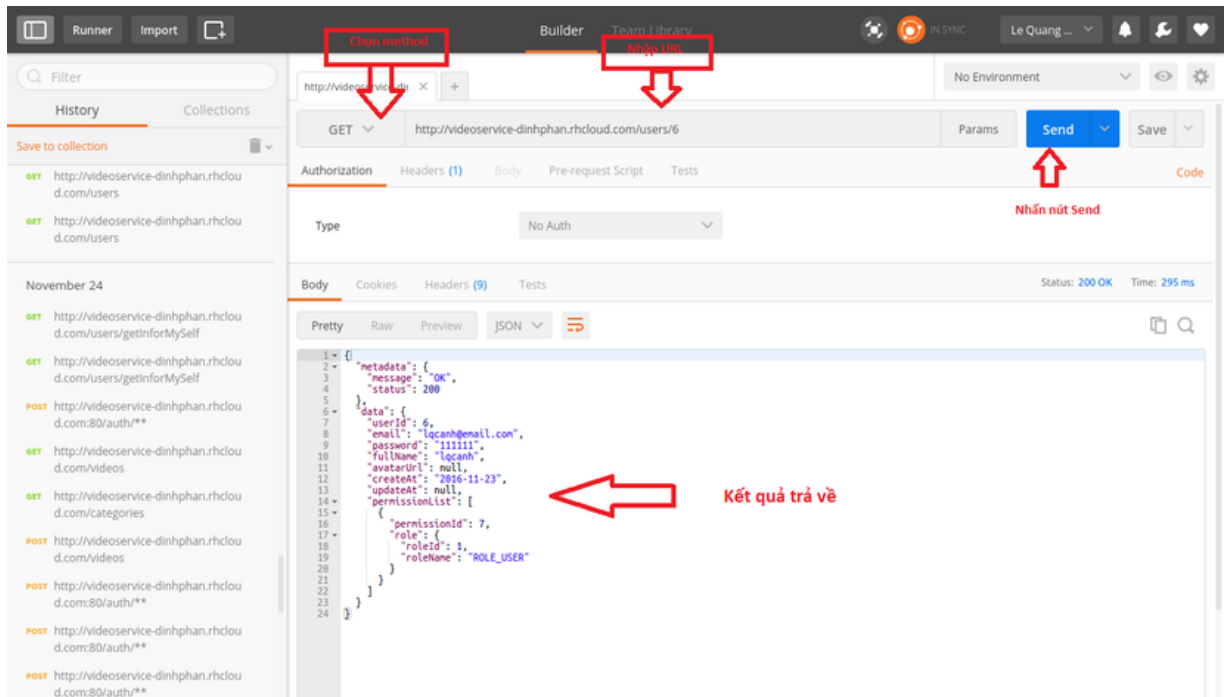
- Cho phép gửi HTTP Request với các phương thức GET, POST, PUT, DELETE.
- Cho phép post dữ liệu dưới dạng form (key-value), text, json
- Hiện kết quả trả về dạng text, hình ảnh, XML, JSON
- Hỗ trợ authorization (Oauth1, 2)
- Cho phép thay đổi header của các request

### Giao diện của Postman

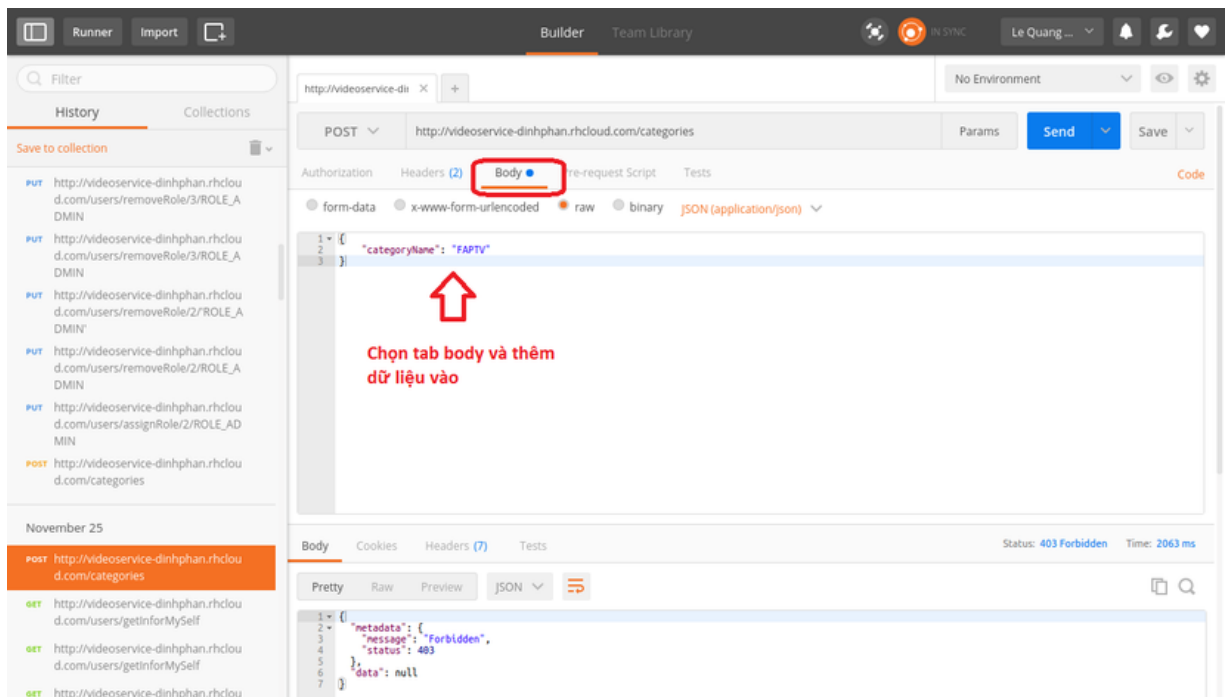


Việc sử dụng POSTMAN rất đơn giản. Bạn chỉ cần chọn method, điền URL, thêm các thông tin cho body, header trong những trường hợp cần thiết, rồi nhấn SEND. Việc của bạn là đợi và POSTMAN sẽ cho bạn kết quả trả về nó có hình thù như thế nào.

### Ví dụ lấy dữ liệu từ Server sử dụng method GET bằng Postman



Method PUT hoặc POST thì cần phải thêm nội dung vào body để gửi request





## **API là gì ?**

API (Application Programming Interface- giao diện lập trình ứng dụng) là một giao diện mà một hệ thống máy tính hay ứng dụng cung cấp để cho phép các yêu cầu dịch vụ có thể được tạo ra từ các chương trình máy tính khác, và/hoặc cho phép dữ liệu có thể được trao đổi qua lại giữa chúng.

Interface ở đây không mang nghĩa giống như UI (User Interface- giao diện người dùng) đâu nhé, ở đây chúng ta nên hiểu nó là một chuẩn/phương pháp để các ứng dụng có thể tương tác, làm việc với nhau.

Hiểu đơn giản thì API là một "con" chuyên đảm nhận một nhiệm vụ duy nhất: đó là xử lý dữ liệu (truy vấn, thêm, sửa, xóa). Còn việc hiển thị ra thì sẽ là nhiệm vụ của một "con" khác, có thể là web, có thể là mobile (android, iOS). Điều này giúp cho ta có thể phát triển ứng dụng trên nhiều nền tảng khác nhau mà chỉ cần đổ dữ liệu ra ngoài giao diện.

## **API làm việc như thế nào ?**

Vậy API làm việc như thế nào? Phía người dùng gửi request, API sẽ gửi lại response là liệu có thể làm được cái người dùng muốn hay ko. Và API được xây dựng trên 2 thành phần chính: Request và Response.

### **Request:**

Một cái request đúng chuẩn cần có 4 thứ:

- URL: là 1 cái địa chỉ duy nhất cho 1 request, thường là đường dẫn tới một hàm xử lý logic.
- Method: là cái hành động người dùng muốn tác động lên dữ liệu. Có 4 loại Method hay được dùng và rất quen thuộc là: GET, POST, PUT, DELETE.
- Headers: nơi chứa các thông tin cần thiết của 1 request nhưng người dùng không biết có sự tồn tại của nó. Ví dụ: độ dài của request body, thời gian gửi request, loại thiết bị đang sử dụng, ...
- Body: nơi chứa thông tin mà người dùng sẽ điền. Giả sử bạn đặt 1 cái bánh pizza, thì thông tin ở phần body sẽ là: Loại bánh pizza, kích cỡ, số lượng đặt.

**Response:**

Sau khi nhận được request từ phía người dùng, API sẽ xử lý cái request đó và gửi ngược lại cho người dùng 1 cái response. Cấu trúc của 1 response tương đối giống phần request nhưng Status code sẽ thay thế cho URL và Method. Tóm lại, nó có cấu trúc 3 phần:

- Status code: là những con số có 3 chữ số và có duy nhất 1 ý nghĩa, ví dụ như vài lỗi quen thuộc “404 Not Found” hoặc “503 Service Unavailable”.
- Headers: giống với headers trong request.
- Body: tương đối giống với trong request.

**Ưu, nhược điểm của Postman:**

Mình cũng mới tìm hiểu nên cũng không dám chắc chắn, đây chỉ là một số ưu, nhược điểm mà mình cảm nhận sau quá trình sử dụng:

**Ưu:**

- Dễ sử dụng, hỗ trợ cả chạy bằng UI và non-UI.
- Hỗ trợ cả RESTful services và SOAP services.
- Có chức năng tạo API document.

**Nhược:**

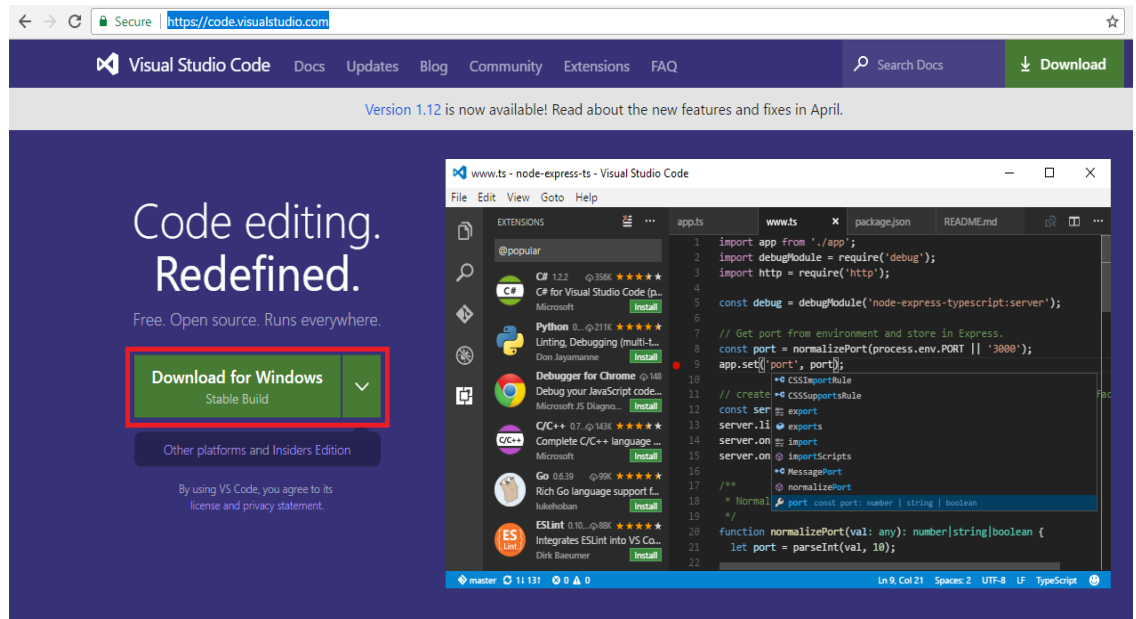
Những bản tính phí mới hỗ trợ những tính năng advance: Làm việc theo team, support trực tiếp...

## II. HƯỚNG DẪN CÀI ĐẶT VÀ SỬ DỤNG

### 1. Cài đặt Visual Studio Code

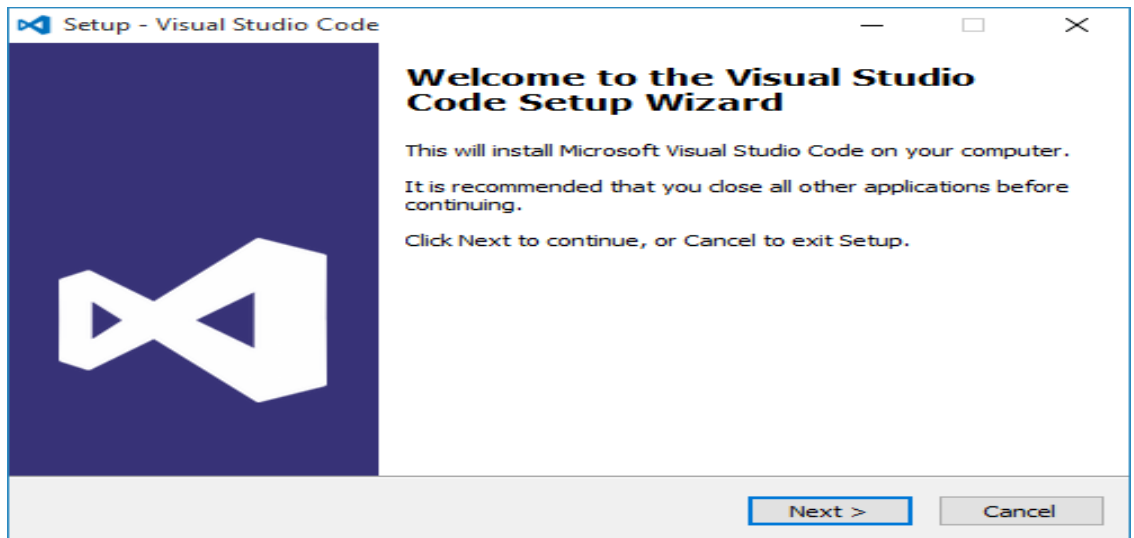
❖ **Bước 1:** Tải Visual Studio Code theo đường dẫn trang chủ:

<https://code.visualstudio.com/>

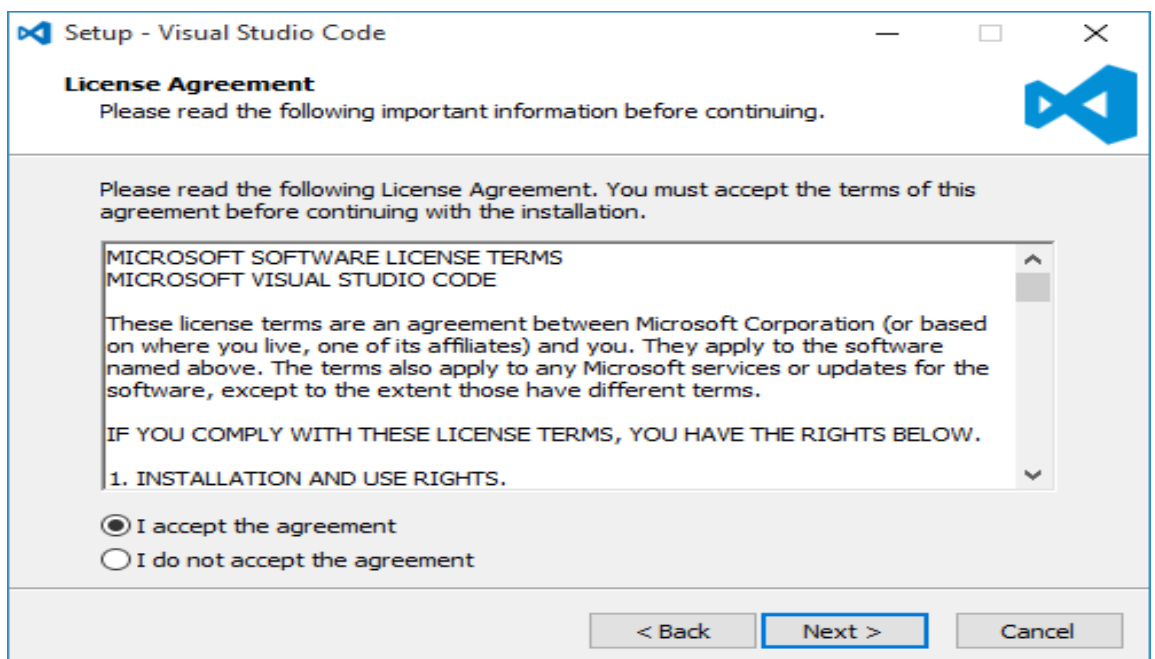


❖ **Bước 2:** Sau khi tải về hoàn tất, chạy file cài đặt.

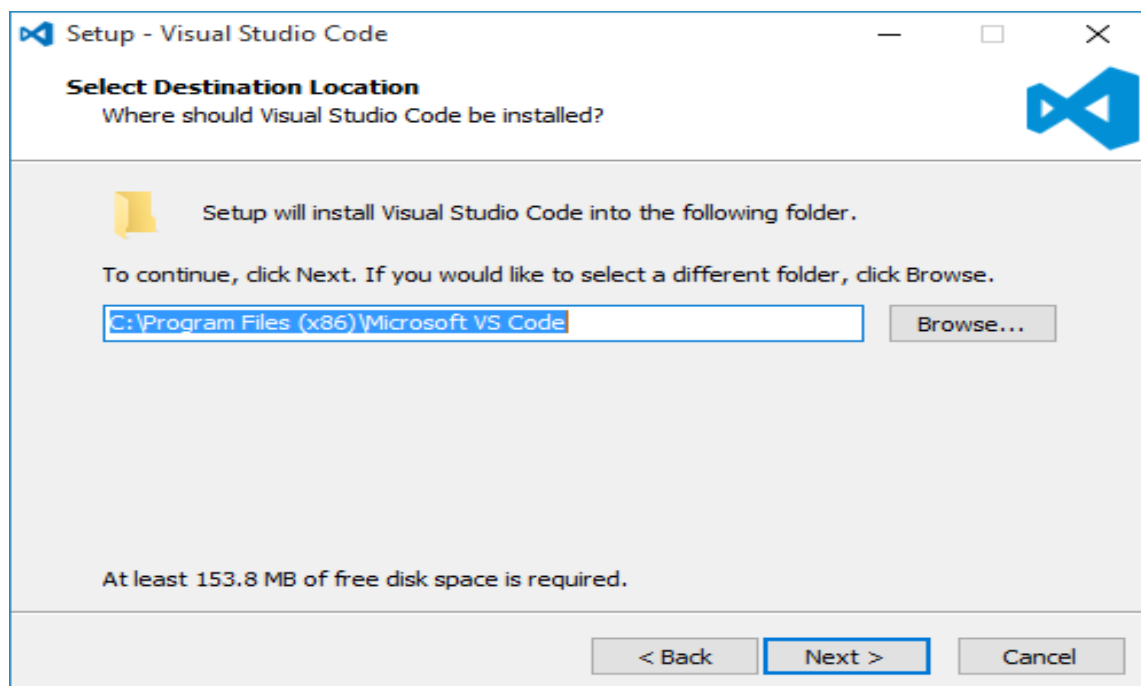
- ❖ **Bước 3:** Trình cài đặt Visual Studio Code xuất hiện, chọn Next để bắt đầu quá trình cài đặt.



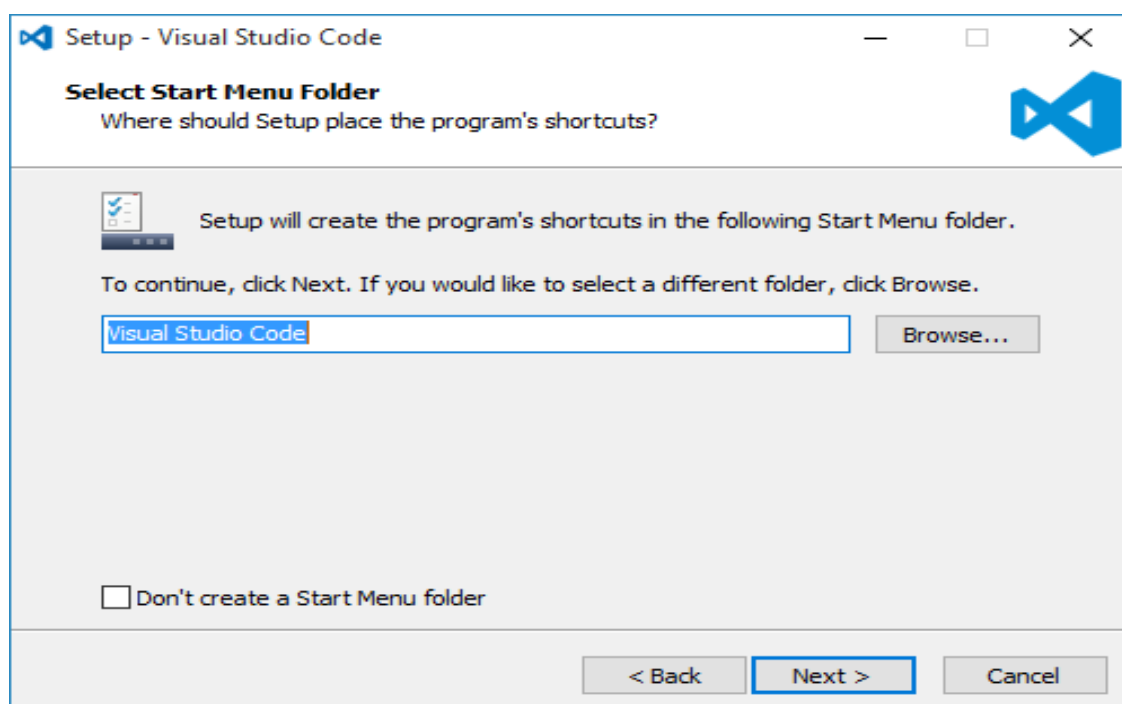
- ❖ **Bước 4:** Chọn đồng ý với các điều khoản của phần mềm và chọn Next để tiếp tục.



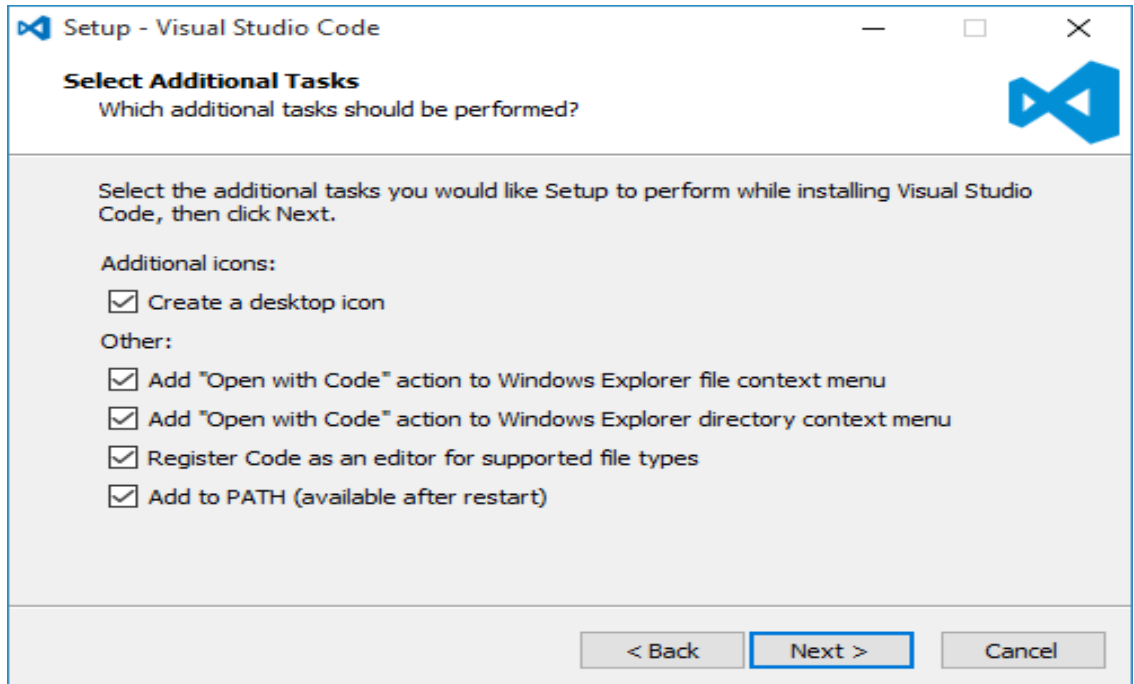
- ❖ **Bước 5:** Tại bước này ta sẽ lựa chọn vị trí cài đặt của Visual Studio Code sau đó chọn Next.



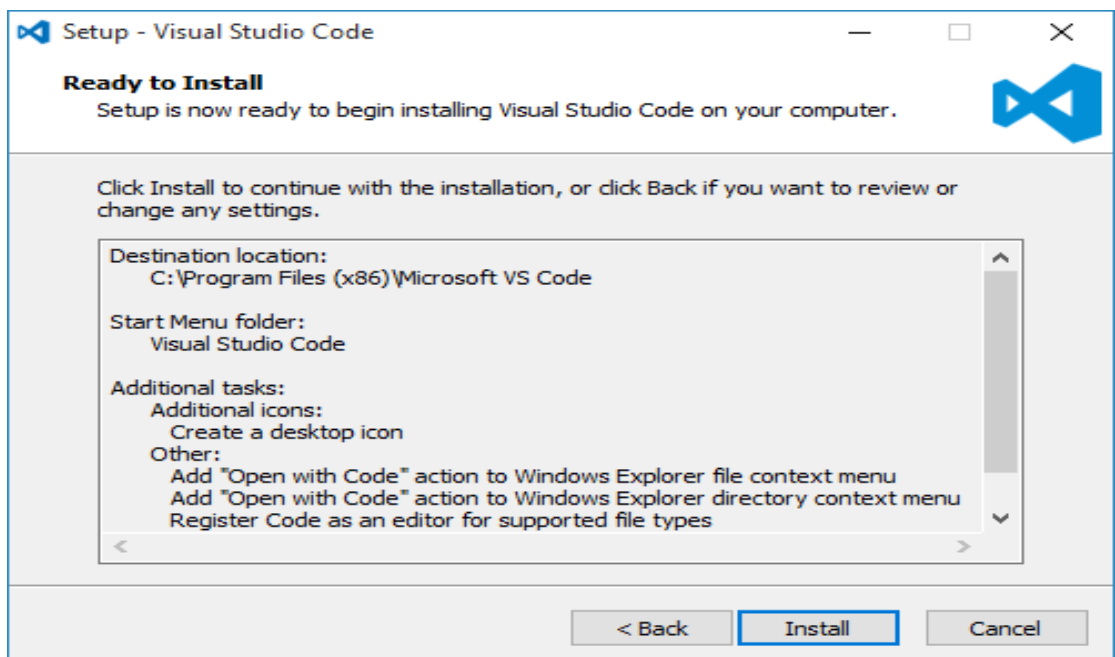
- ❖ **Bước 6:** Cho phép Visual Studio Code tạo Menu Folder và chọn Next để tiếp tục.



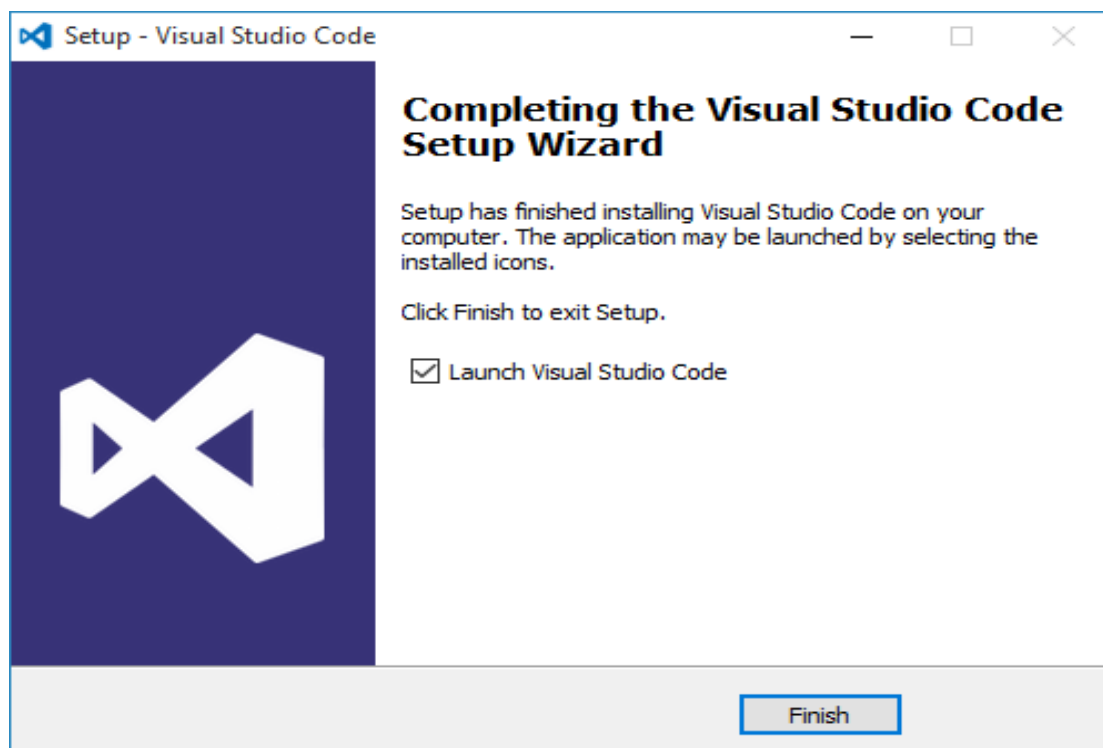
- ❖ **Bước 7:** Các lựa chọn cài đặt thêm, chọn Next để tiếp tục.



- ❖ **Bước 8:** Xác nhận lần cuối các lựa chọn trước khi bắt đầu quá trình cài đặt. Chọn Install để bắt đầu quá trình cài đặt.

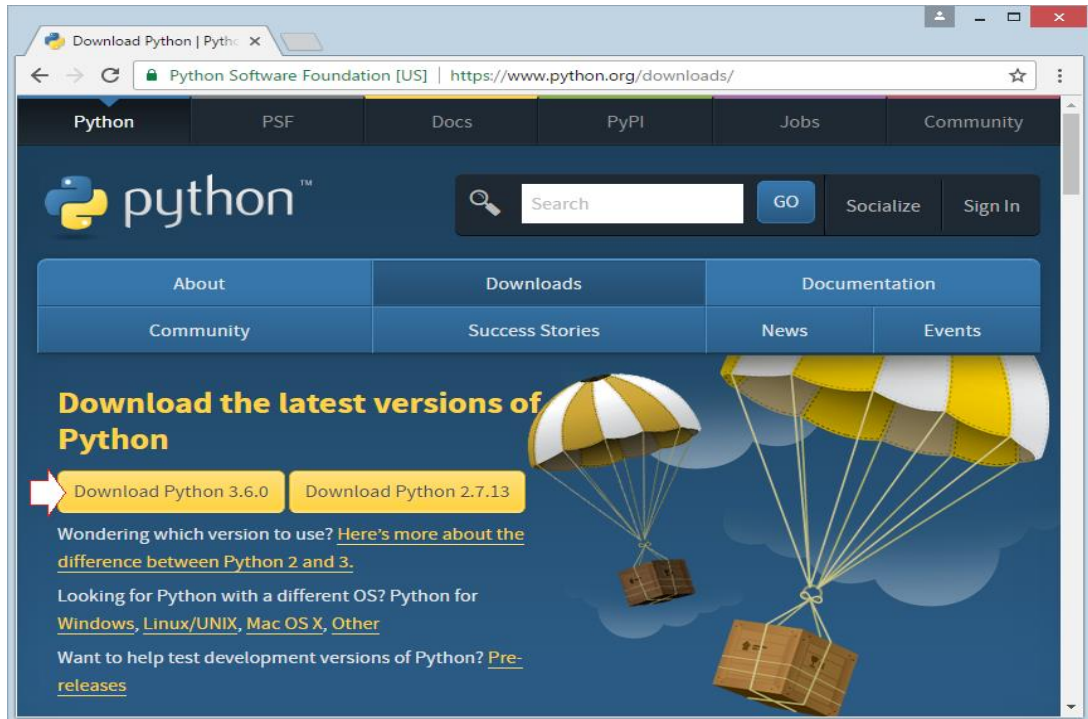


- ❖ **Bước 9:** Chọn Finish để hoàn tất quá trình cài đặt, ta cho phép Visual Studio Code chạy sau khi hộp thoại này được đóng lại.

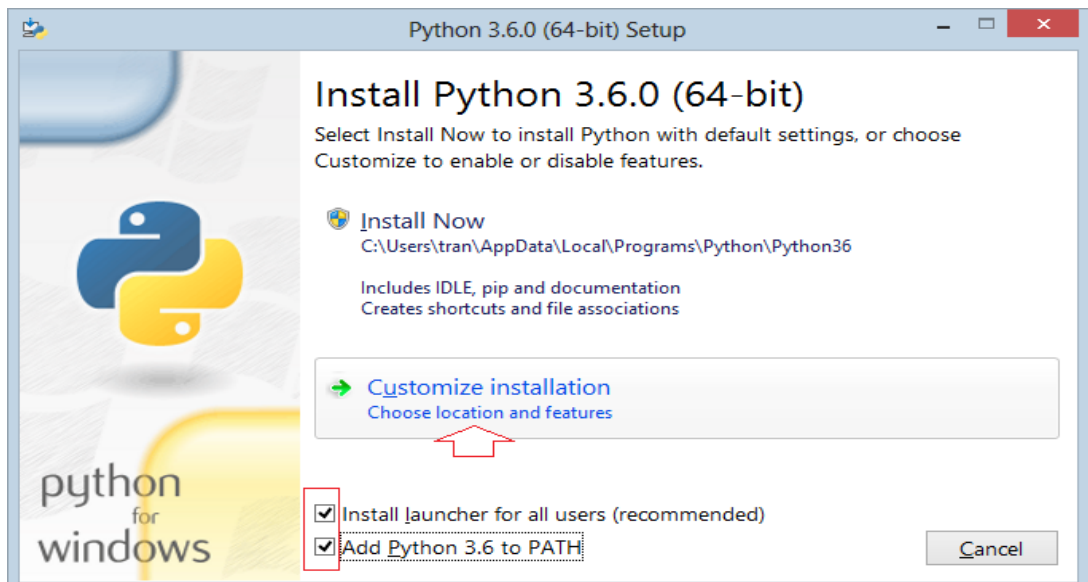


## 2. Cài đặt Python

- ❖ **Bước 1:** Tải trình cài đặt Python từ trang chủ theo đường dẫn: <https://www.python.org/downloads/>

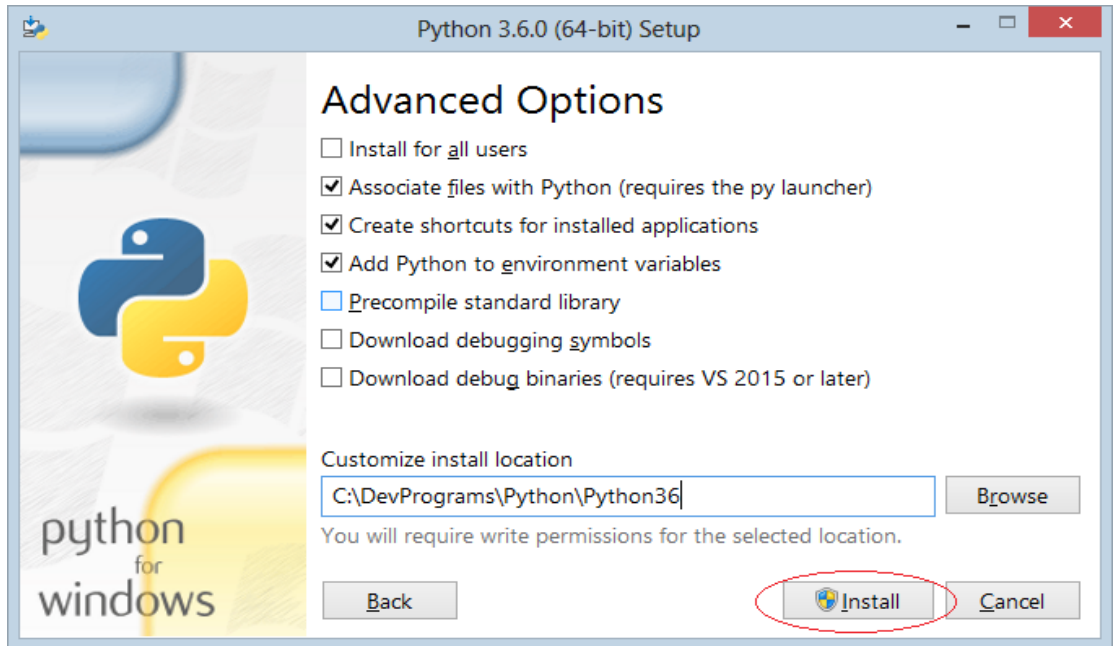


- ❖ **Bước 2:** Cài đặt python bằng file vừa tải về ở bước 1, Chọn "Customize Installation" để bạn có thể tùy chọn vị trí Python sẽ được cài đặt ra.

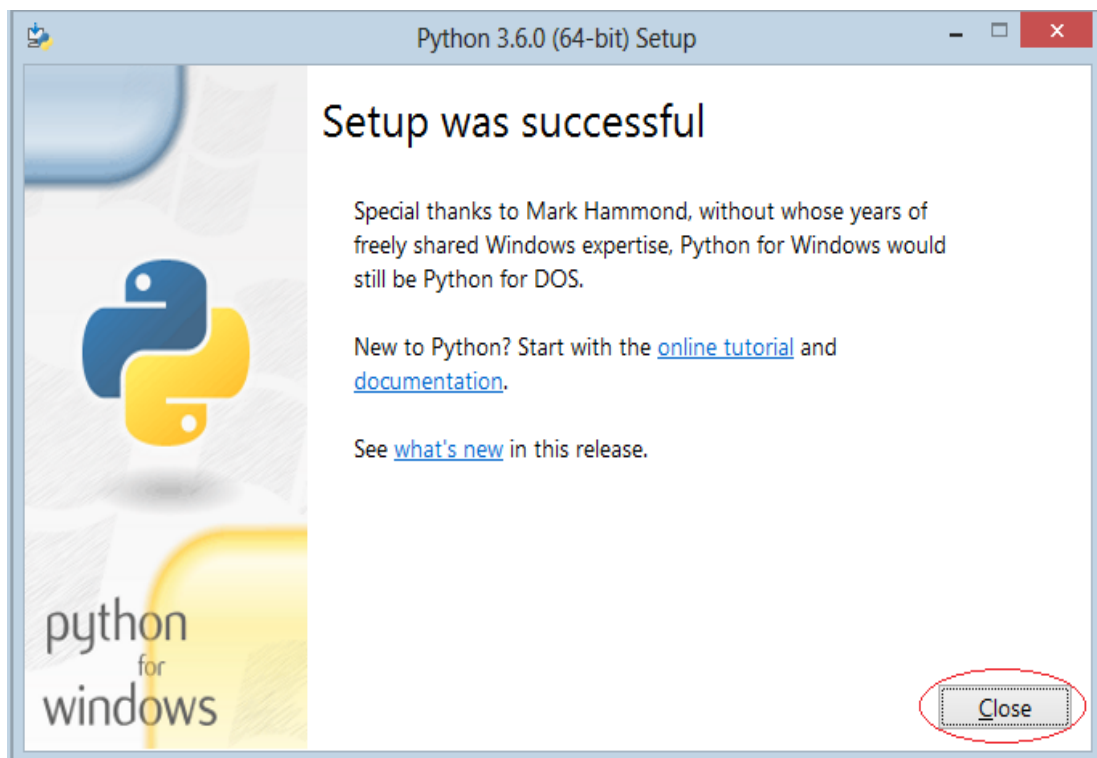


- ❖ **Bước 3:** Chọn vị trí mà Python sẽ được cài đặt ra.



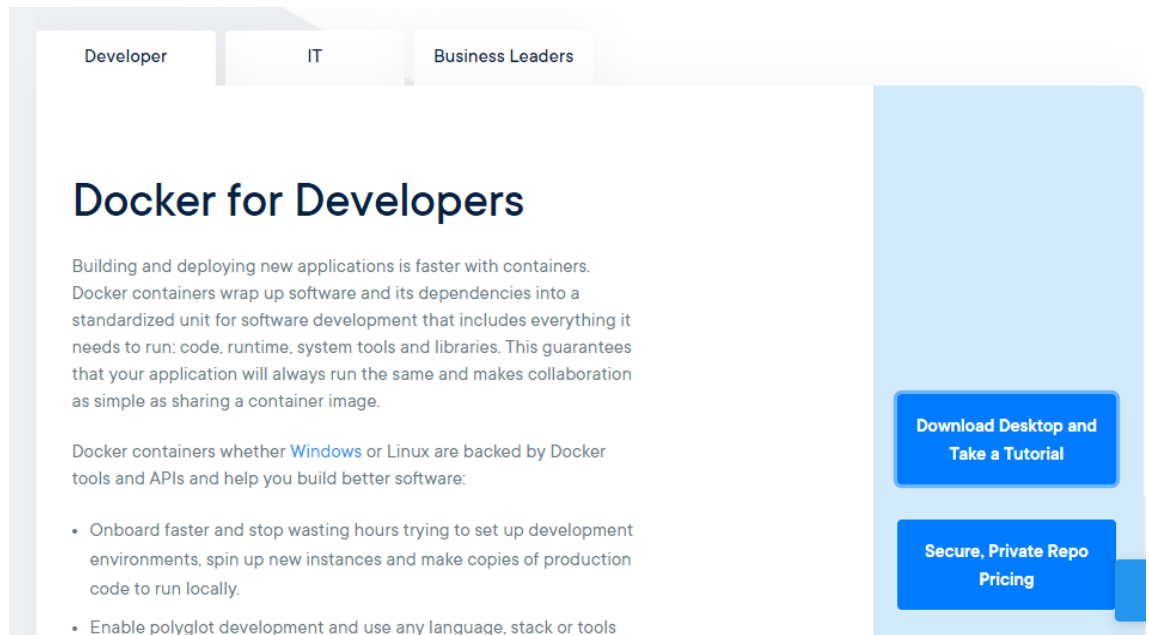


❖ **Bước 4:** Chọn “Install” và đợi quá trình cài đặt hoàn tất.

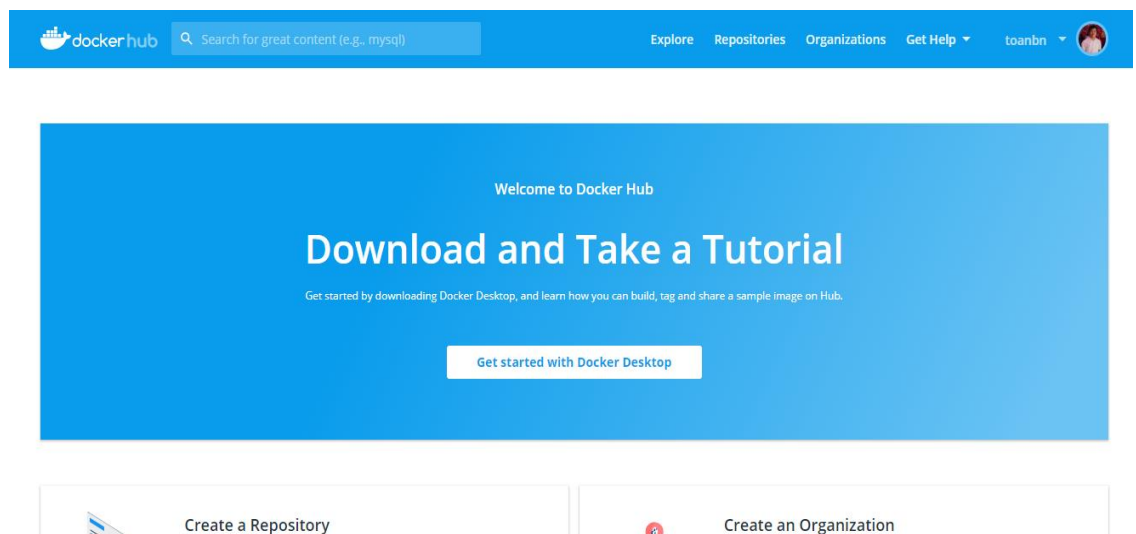


### 3. Cài đặt Docker

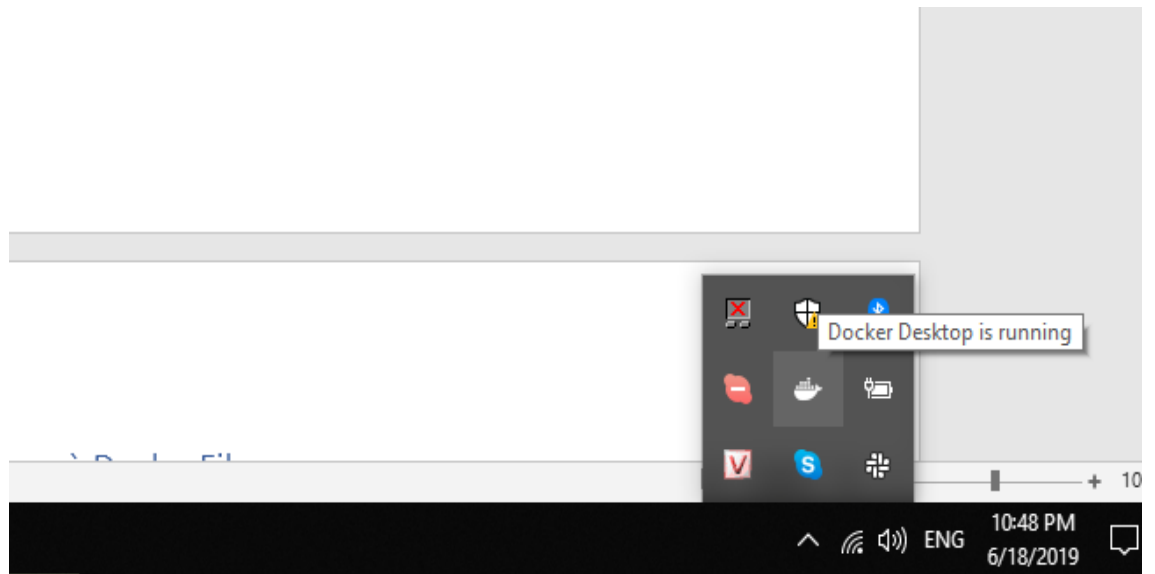
❖ **Bước 1:** Ban đầu chúng ta vào <https://www.docker.com/> và chọn Get Started, các bạn sẽ vào <https://www.docker.com/get-started>. Sau đó các bạn sẽ thấy phần download Docker cho Developer



- ❖ **Bước 2:** Bạn cần có tài khoản để download bộ cài. Sau khi download bạn cứ cài đặt như bình thường. Sau khi đăng nhập bạn sẽ vào <https://hub.docker.com/> để download bộ cài và có thể tạo repository trong này để lưu image



- ❖ **Bước 3:** Sau khi cài xong bạn có thể khởi động nó ở Start với ứng dụng Docker for Desktop. Và nếu chạy thành công thì bạn sẽ có biểu tượng Docker is Running ở Tray Icon



#### 4. Cài đặt thư viện spaCy

- ❖ **Bước 1:** Cài đặt Python vào máy tính. (đã cài đặt)
- ❖ **Bước 2:** Cài đặt Visual Studio Code. (đã cài đặt)
- ❖ **Bước 3:** Mở Visual Studio Code, Sử dụng tổ hợp phím “Ctrl + ~” mở bảng Command Line.
- ❖ **Bước 4:** Nhập lệnh “pip install spaCy” để cài đặt spaCy.

```
$ pip install -U spacy
```

- ❖ **Bước 5:** Sau khi quá trình cài đặt hoàn tất, màn hình Command Line hiện như hình ảnh dưới đây là quá trình cài đặt thành công.

```
(.env) C:\Windows\system32>pip install spacy --log log.txt
Collecting spacy
  Using cached https://files.pythonhosted.org/packages/28/52/8d21458ae9345d8480b11d33d66829795b8069cabf223d2f9f99da0d75ad/spacy-2.0.16-cp36-cp36m-win_amd64.whl
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\windows\system32\env\lib\site-packages (from spacy) (1.0.1)
Requirement already satisfied: msgpack-numpy<0.4.4 in c:\windows\system32\env\lib\site-packages (from spacy) (0.4.3.2)
Requirement already satisfied: plac<1.0.0,>=0.9.6 in c:\windows\system32\env\lib\site-packages (from spacy) (0.9.6)
Requirement already satisfied: ujson>=1.35 in c:\windows\system32\env\lib\site-packages (from spacy) (1.35)
Requirement already satisfied: numpy>=1.15.0 in c:\windows\system32\env\lib\site-packages (from spacy) (1.15.4)
Requirement already satisfied: preshed<2.1.0,>=2.0.1 in c:\windows\system32\env\lib\site-packages (from spacy) (2.0.1)
Requirement already satisfied: regex==2018.01.10 in c:\windows\system32\env\lib\site-packages (from spacy) (2018.1.10)
Requirement already satisfied: thinc<6.13.0,>=6.12.0 in c:\windows\system32\env\lib\site-packages (from spacy) (6.12.0)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\windows\system32\env\lib\site-packages (from spacy) (2.20.1)
Requirement already satisfied: dill<0.3,>=0.2 in c:\windows\system32\env\lib\site-packages (from spacy) (0.2.8.2)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\windows\system32\env\lib\site-packages (from spacy) (2.0.2)
Requirement already satisfied: msgpack>=0.3.0 in c:\windows\system32\env\lib\site-packages (from msgpack-numpy<0.4.4->spacy) (0.5.6)
Requirement already satisfied: six<2.0.0,>=1.10.0 in c:\windows\system32\env\lib\site-packages (from thinc<6.13.0,>=6.12.0->spacy) (1.11.0)
Requirement already satisfied: tqdm<5.0.0,>=4.10.0 in c:\windows\system32\env\lib\site-packages (from thinc<6.13.0,>=6.12.0->spacy) (4.28.1)
Requirement already satisfied: wrapt<1.11.0,>=1.10.0 in c:\windows\system32\env\lib\site-packages (from thinc<6.13.0,>=6.12.0->spacy) (1.10.11)
Requirement already satisfied: cytoolz<0.10,>=0.9.0 in c:\windows\system32\env\lib\site-packages (from thinc<6.13.0,>=6.12.0->spacy) (0.9.0.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\windows\system32\env\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (2018.10.15)
Requirement already satisfied: idna<2.8,>=2.5 in c:\windows\system32\env\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (2.7)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\windows\system32\env\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (3.0.4)
Requirement already satisfied: urllib3<1.25,>=1.21.1 in c:\windows\system32\env\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (1.24.1)
Requirement already satisfied: pyreadline>=1.7.1 in c:\windows\system32\env\lib\site-packages (from dill<0.3,>=0.2->spacy) (2.1)
Requirement already satisfied: toolz>=0.8.0 in c:\windows\system32\env\lib\site-packages (from cytoolz<0.10,>=0.9.0->thinc<6.13.0,>=6.12.0->spacy) (0.9.0)
Installing collected packages: spacy
Successfully installed spacy-2.0.16
```

## 5. Cài đặt gói nhận dạng ngôn ngữ tiếng Việt cho spaCy

- ❖ **Bước 1:** Cài đặt thư viện “pyvi” bằng Command Line : “pip install pyvi”
- ❖ **Bước 2:** Tải gói nhận dạng ngôn ngữ tiếng Việt từ github bằng Command Line :

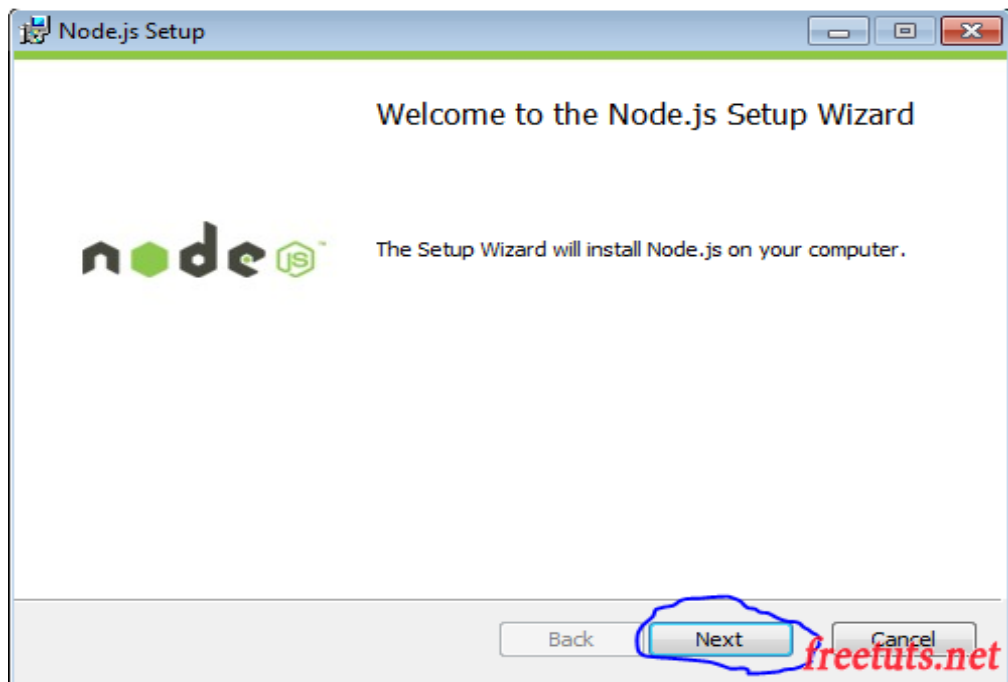
*pip install*

[https://github.com/trungtv/vi\\_spacy/raw/master/packages/vi\\_spacy\\_model-0.2.1/dist/vi\\_spacy\\_model-0.2.1.tar.gz](https://github.com/trungtv/vi_spacy/raw/master/packages/vi_spacy_model-0.2.1/dist/vi_spacy_model-0.2.1.tar.gz)

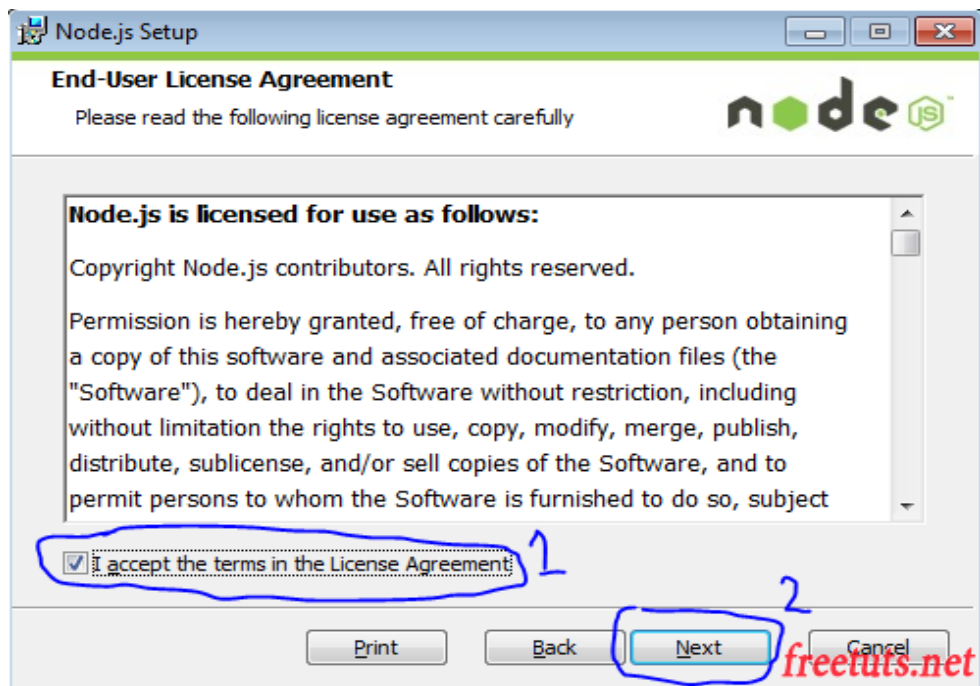
## 6. Cài đặt Nodejs

- ❖ **Bước 1:** Tải file cài đặt từ trang chủ của Node: <http://nodejs.org/>

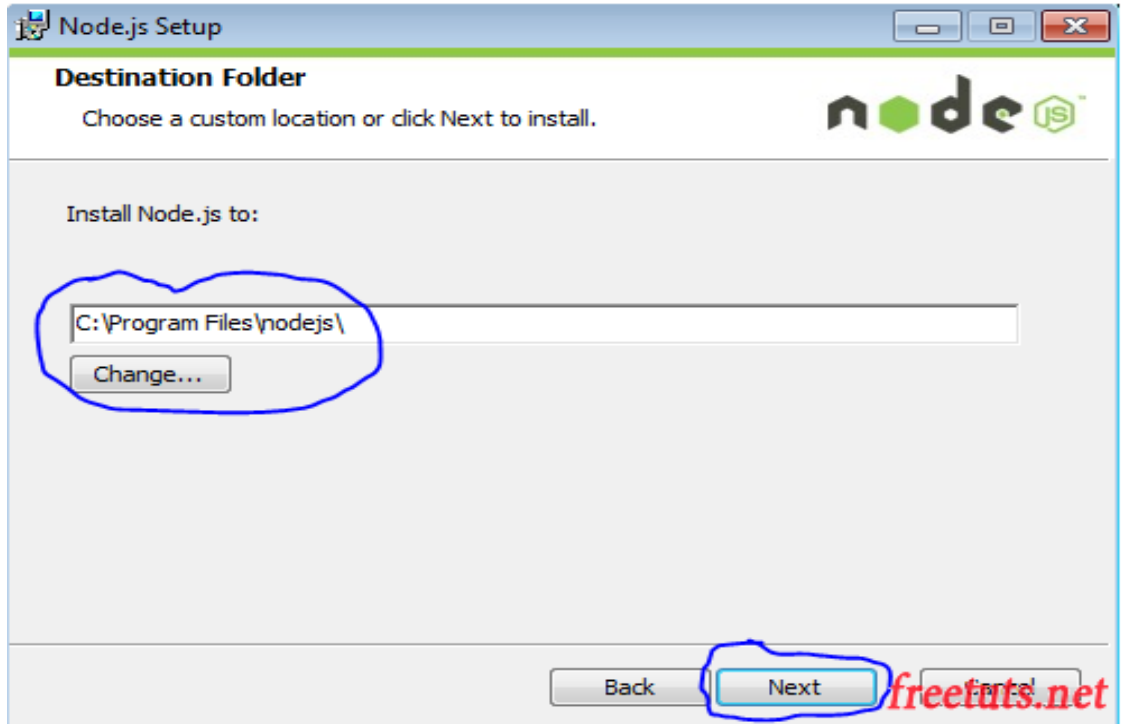
- ❖ **Bước 2:** Cài đặt phần mềm Nodejs bằng cách chạy file cài đặt tải về từ bước trên.



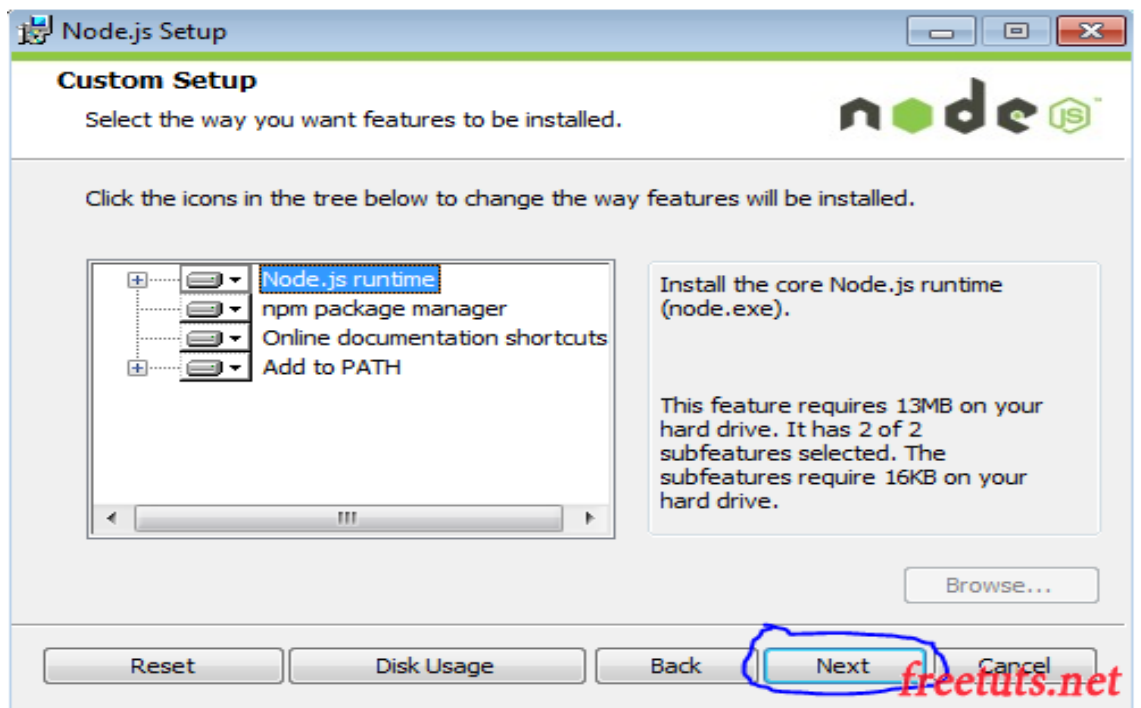
- ❖ **Bước 3:** Bạn check vào ô I accept the terms in the License Agreement và click Next.



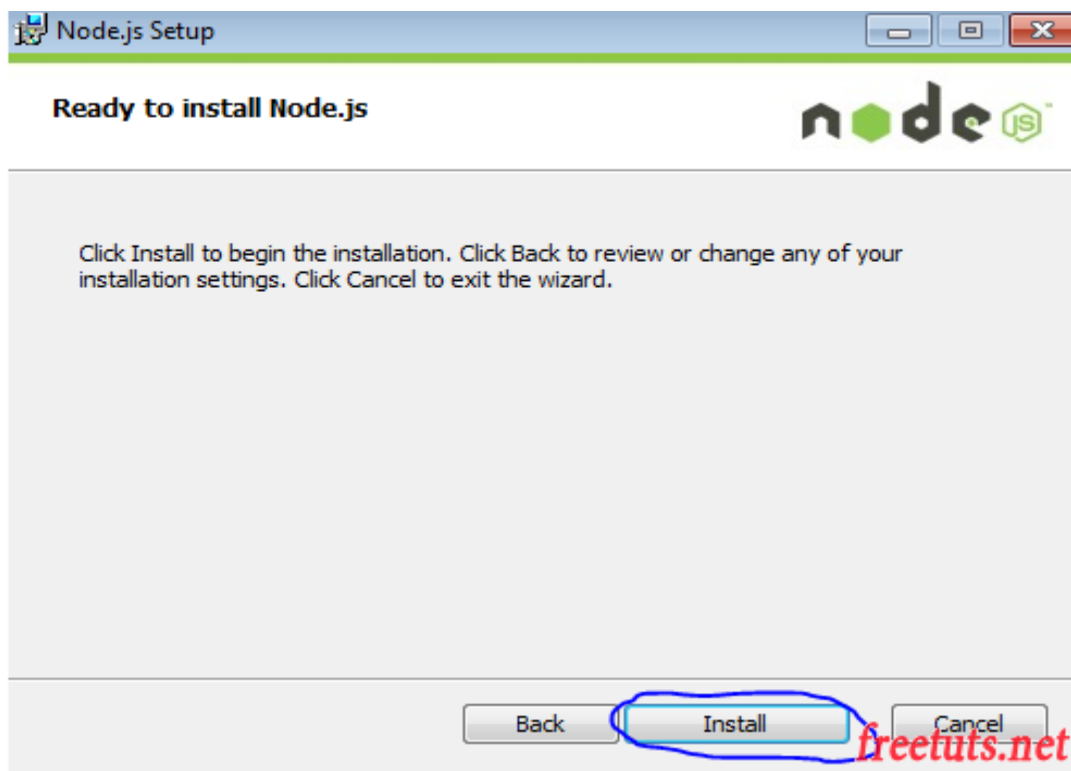
- ❖ **Bước 4:** Giao diện mới xuất hiện, bạn chọn đường dẫn lưu và click Next.



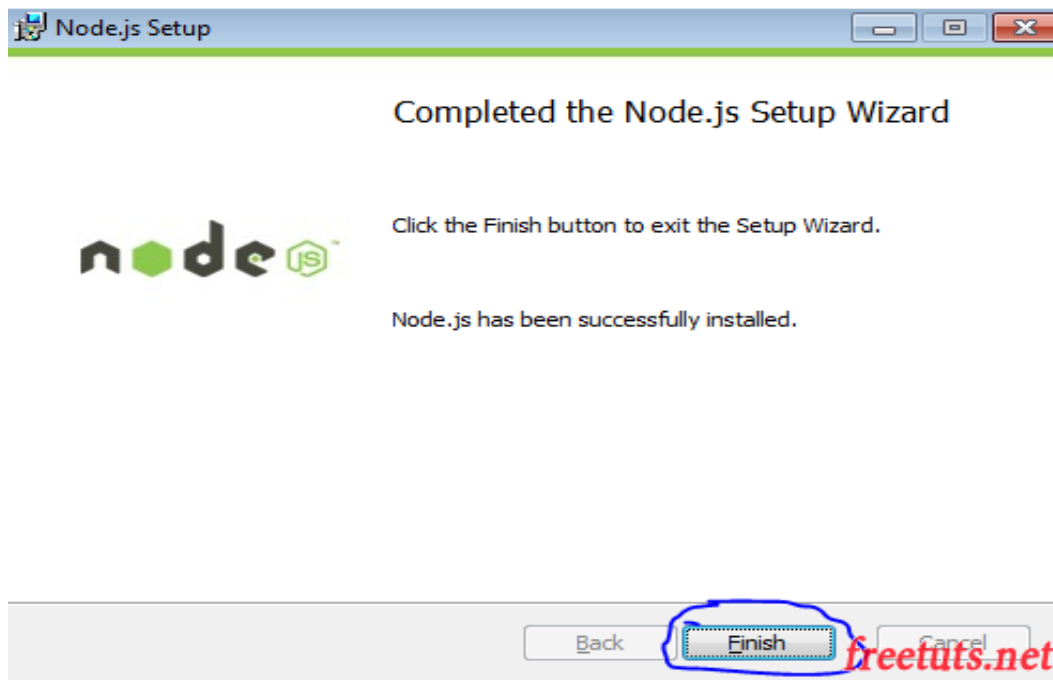
- ❖ **Bước 5:** Giao diện mới xuất hiện, bạn để nguyên NodeJS lựa chọn các gói mặc định, click vào nút Next.



- ❖ **Bước 6:** Tiếp tục click vào nút Install để cài đặt.



- ❖ **Bước 7:** Công đoạn cài đặt sẽ mất khoảng một phút đồ lại nên bạn vui lòng chờ nó chạy xong và click vào nút Finish.



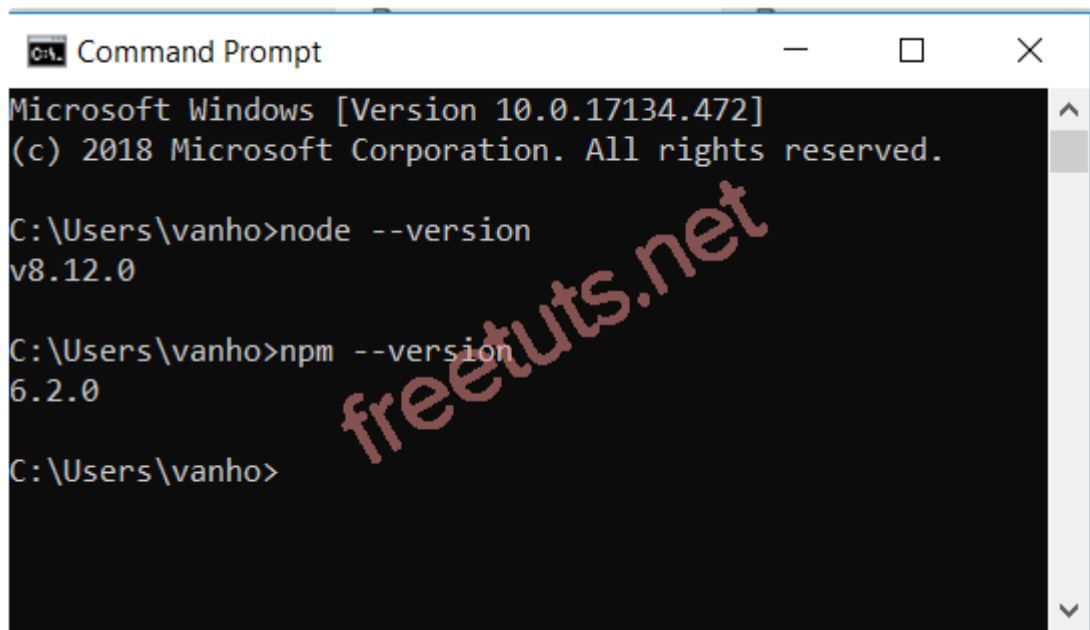
## 7. Cài đặt Nodejs express

- ❖ **Bước 1:** Cài đặt Node (đã cài đặt.)

- ❖ **Bước 2:** Kiểm tra NodeJs và npm đã được cài hay chưa bạn hãy mở cmd dùng hai lệnh dưới đây:

```
1 | node --version
2 | npm --version
```

Nếu bạn nhận được kết quả như hình dưới thì bạn đã cài đặt thành công NodeJs và npm rồi đấy.



```
Microsoft Windows [Version 10.0.17134.472]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\vanho>node --version
v8.12.0

C:\Users\vanho>npm --version
6.2.0

C:\Users\vanho>
```

- ❖ **Bước 3:** Dùng npm để cài đặt ExpressJs. Mở CommandLine và di chuyển đến thư mục dự án muốn cài đặt, và thực hiện câu lệnh sau:

```
1 | npm init
```

- ❖ **Bước 4:** Tới đây là bạn mới chỉ tạo được khung sườn cho dự án chứ chưa cài đặt Express, bạn cần chạy thêm lệnh sau để bắt đầu cài đặt.

```
1 | npm install --save express
```

## 8. Cài đặt thư viện Flask

- ❖ **Bước 1:** Cài đặt Python (đã cài đặt)
- ❖ **Bước 2:** Cài đặt Visual Studio Code (đã cài đặt)
- ❖ **Bước 3:** Mở Visual Studio Code, Sử dụng tổ hợp phím “Ctrl + ~” mở bảng Command Line.



- ❖ **Bước 4:** Sử dụng câu lệnh bên dưới để tiến hành cài đặt Flask vào máy tính.

```
$ pip install Flask
```

## 9. Cài đặt MongoDB trong Docker

- ❖ **Bước 1:** Cài đặt Docker (đã cài đặt)
- ❖ **Bước 2:** Mở CommandLine và sử dụng câu lệnh dưới đây để tiến hành tải xuống và cài đặt MongoDB trong Docker.

```
$ docker run --name some-mongo -d mongo:tag
```

- ❖ **Bước 3:** Kiểm tra Mongo đã cài thành công hay chưa bằng cách nhập câu lệnh “docker ps” vào CommandLine, nếu xuất hiện kết quả như hình dưới thì đã cài đặt thành công.

```
eric-burel@eric-ThinkPad-T450s:~$ docker run --name mymongodb -d mongo
df761c06d9adfe768f1c0d320d4dfc13f9f557551553053a8bee89757162652
eric-burel@eric-ThinkPad-T450s:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
df761c06d9ad	mongo	"docker-entrypoint.s..."	2 seconds ago	Up 2 seconds
5de633c3f97e	mongo	"docker-entrypoint.s..."	33 minutes ago	Up 21 minutes
0.0.0.0:27017->27017/tcp	my-awesome-db			

## 10. Cài đặt Ghost CMS

- ❖ **Bước 1:** Cài đặt Docker (đã cài đặt)
- ❖ **Bước 2:** Mở CommandLine và sử dụng câu lệnh dưới đây để tiến hành tải xuống và cài đặt MongoDB trong Docker.

```
$ docker run -d --name some-ghost ghost
```

Mặc định Ghost trong Docker sẽ chạy ở cổng “2368”.

## 11. Cài đặt SpiderKeeper

- ❖ **Bước 1:** Cài đặt Docker (đã cài đặt)
- ❖ **Bước 2:** Mở CommandLine và sử dụng câu lệnh dưới đây để tiến hành tải xuống và cài đặt MongoDB trong Docker.

```
pip install spiderkeeper
```

## 12. Cài đặt Scrappy

- ❖ **Bước 1:** Cài đặt Python (đã cài đặt)
- ❖ **Bước 2:** Cài đặt Visual Studio Code (đã cài đặt)
- ❖ **Bước 3:** Mở Visual Studio Code, Sử dụng tổ hợp phím “Ctrl + ~” mở bảng Command Line.
- ❖ **Bước 4:** Sử dụng câu lệnh bên dưới để tiến hành cài đặt Flask vào máy tính.

```
pip install scrapy
```