# SUMMER INTERNSHIP REPORT

## NATIONAL INSTITUTE OF TECHNOLOGY, DELHI


Submitted By :

**Yash n Chaudhari**

**181210062**

**B. Tech CSE**

Submitted To :

**Dr. Rajya Lakshmi**

**Department of CSE**

**NIT DELHI**

# Acknowledgement

I could not have done this work without the help I received cheerfully from whole NIT Delhi. I would like to thank Dr. Rajya Lakshmi Lella for providing me the opportunity to work on this project for suggesting the nice ideas to work upon.The internship provided me an opportunity to learn **Python** and put my skills to the test all the while helping me to enhance my knowledge in the strict and method process of developing software, exploring libraries and framework, proper documentation, and multi-threading. I am also highly indebted to my friends who helped in reviewing the program and testing during the development phase, providing feedback on the working and suggestions for improvement. I consider this opportunity as a substantial milestone in my career that I desire to see grow. I shall keep exploring, and improving with the skills and experience that I have obtained.

# Certificate of Internship

राष्ट्रीयप्रौद्योगिकीसंस्थानदिल्ली
NATIONAL INSTITUTE OF TECHNOLOGY DELHI
(An autonomous Institute under the aegis of Ministry of HRD, Govt. of India)
सेक्टर ए-७, इन्स्टिटूशनल एरिया, नरेला, दिल्ली-११००४०,भारत/Sector A-7, Institutional AreaNarela, Delhi-110040,
INDIAदूरभाष/Tele: +9111-27787500-503, फैक्स/ Fax: +9111-27787503
वेबसाइट/Website: www.nitdelhi.ac.in

---

**SUMMER INTERNSHIP**

**Mr. Yash Chaudhari (181210062),** B. Tech 3rd year student of Computer Science and Engineering (CSE) Department, National Institute of Technology Delhi has successfully completed his internship program (during summer after his semester IV), on a Project Titled "**Study Material Download and Intelligent Phrase Identification Program**" from **May 23, 2020, to July 31, 2020** under my supervision and has obtained satisfactory results.

During his internship, he was found to be hardworking and sincere.

L. Rajya Lakshmi

**(L. Rajya Lakshmi)**

# Contents

# 1. Python

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was created in the late 1980s, and first released in 1991, by Guido van Rossum as a successor to the ABC programming language. Python 2.0, released in 2000, introduced new features, such as list comprehensions, and a garbage collection system with reference counting, and was discontinued with version 2.7 in 2020.

Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible and much Python 2 code does not run unmodified on Python 3.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, a free and open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development. It currently ties with Java as the second most popular programming language in the world.

# 1.1 History

Python was conceived in the late 1980s by **Guido van Rossum** at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC programming language, which was inspired by SETL), capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989.

Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's Benevolent Dictator For Life, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. He now shares his leadership as a member of a five-person steering council.In January 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing and Van Rossum to a five-member "Steering Council" to lead the project. Guido van Rossum has since then withdrawn his nomination for the 2020 Steering council.

Python 2.0 was released on 16 October 2000 with many major new features, including a cycle-detecting garbage collector and support for Unicode.

Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible. Many of its major features were backported to Python 2.6.x and 2.7.x version series. Releases of Python 3 include the 2to3 utility, which automates (at least partially) the translation of Python 2 code to Python 3.

Python 2.7's end-of-life date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to

Python 3. No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.6.x and later are supported.

## 1.2 Design philosophy

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's design offers some support for functional programming in the Lisp tradition. It has filter, map, and reduce functions; list comprehensions, dictionaries, sets, and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

he language's core philosophy is summarized in the document The Zen of Python, which includes aphorisms such as:

- Beautiful is better than ugly.
- Explicit is better than implicit.

- Simple is better than complex.
- Complex is better than complicated.
- Readability counts.

## 1.3 Features

Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of the CPython reference implementation that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name as a tribute to the British comedy group Monty Python and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar.

A common neologism in the Python community is pythonic, which can have a wide range of meanings related to program style. To say that code is pythonic is to say that it uses Python idioms well, that it is natural or shows fluency in the language, that it conforms with Python's minimalist philosophy and emphasis on readability. In contrast, code that is difficult to understand or reads like a rough transcription from another programming language is called unpythonic.

## 1.4 Languages Influenced

- Boo
- Cobra
- CoffeeScript
- ECMAScript/JavaScript
- GDScript
- Go
- Groovy
- Julia
- Nim
- Ruby
- Swift

# 2. Abstract

The idea behind the project is to help students acquire study material from a specific website without having them navigate to it avoiding any tedious work and it may also help students to learn how a specific website is designed and programmed by studying the files used in the making of the website.

The program automatically segregates the files into specific folders respective to their extensions. The program stores the files where the program itself is located.

The program is designed to search all the pdf files for a specific phrase entered by the user and show you the five PDF files which are highly relevant to the phrase entered by the user. So that they won't have to go through all of it.

The program shows related words to the phrase entered so that the user may research those phrases and gain more knowledge in the Subject.

# 3. Aim

To Learn Python and create an automated Python project that does the following tasks :

**1.** Parses a given Website and identifies different kinds of files attached to that Website.

**2.** Extracts the files identified.

**3.** Creates a separate folder for each file type.

**4.** Stores the extracted files in their respective folders.

**5.** Searches all the PDF files downloaded for the given phrase.

**6.** Displays 5 PDF files that are highly relevant to a given phrase in the files downloaded.

**7.** Identifies and searches other phrases related to the given phrase.

# 4. Technology Used

## 4.1 Automation

Automation is the technology by which a process or procedure is performed with minimal human assistance. Automation,or automatic control, is the use of various control systems for operating equipment such as machinery, processes in factories, boilers, and heat-treating ovens, switching on telephone networks, steering, and stabilization of ships, aircraft, and other applications and vehicles with minimal or reduced human intervention.

Automation covers applications ranging from a household thermostat controlling a boiler, to a large industrial control system with tens of thousands of input measurements and output control signals. In control complexity, it can range from simple on-off control to multi-variable high-level algorithms.

In the simplest type of an automatic control loop, a controller compares a measured value of a process with a desired set value, and processes the resulting error signal to change some input to the process, in such a way that the process stays at its set point despite disturbances. This closed-loop control is an application of negative feedback to a system. The mathematical basis of control theory was begun in the 18th century and advanced rapidly in the 20th.

Automation has been achieved by various means including mechanical, hydraulic, pneumatic, electrical, electronic devices, and computers, usually in combination. Complicated systems, such as modern factories, airplanes, and ships typically use

all these combined techniques. The benefit of automation includes labor savings, savings in electricity costs, savings in material costs, and improvements to quality, accuracy, and precision.

The World Bank's World Development Report 2019 shows evidence that the new industries and jobs in the technology sector outweigh the economic effects of workers being displaced by automation.

The term automation, inspired by the earlier word automatic (coming from automaton), was not widely used before 1947, when Ford established an automation department. It was during this time that industry was rapidly adopting feedback controllers, which were introduced in the 1930s.

## 4.2 Web Scraping

Web scraping is the process of using bots to extract content and data from a website.Unlike screen scraping, which only copies pixels displayed on screen, web scraping extracts underlying HTML code and, with it, data stored in a database. The scraper can then replicate entire website content elsewhere.

Web scraping is used in a variety of digital businesses that rely on data harvesting. Legitimate use cases include:

Search engine bots crawling a site, analyzing its content and then ranking it.

Price comparison sites deploying bots to auto-fetch prices and product descriptions for allied seller websites.

Market research companies using scrapers to pull data from forums and social

media (e.g., for sentiment analysis).

Web scraping is also used for illegal purposes, including the undercutting of prices and the theft of copyrighted content. An online entity targeted by a scraper can suffer severe financial losses, especially if it's a business strongly relying on competitive pricing models or deals in content distribution.

Content scraping comprises large-scale content theft from a given site. Typical targets include online product catalogs and websites relying on digital content to drive business. For these enterprises, a content scraping attack can be devastating.

## 4.3 Multithreading

Multithreading is the ability of a central processing unit (CPU) (or a single core in a multi-core processor) to provide multiple threads of execution concurrently, supported by the operating system. This approach differs from multiprocessing. In a multithreaded application, the threads share the resources of a single or multiple cores, which include the computing units, the CPU caches, and the translation lookaside buffer (TLB).

Where multiprocessing systems include multiple complete processing units in one or more cores, multithreading aims to increase utilization of a single core by using thread-level parallelism, as well as instruction-level parallelism. As the two techniques are complementary, they are sometimes combined in systems with multiple multithreading CPUs and with CPUs with multiple multithreading cores.

The multithreading paradigm has become more popular as efforts to further

exploit instruction-level parallelism have stalled since the late 1990s. This allowed the concept of throughput computing to re-emerge from the more specialized field of transaction processing.

Even though it is very difficult to further speed up a single thread or single program, most computer systems are actually multitasking among multiple threads or programs. Thus, techniques that improve the throughput of all tasks result in overall performance gains.

Two major techniques for throughput computing are multithreading and multiprocessing.

# 5. Approach

## 5.1 Web Scraping

The program initially needed to identify the files to be downloaded , as these files could be of any type , that is it may have any extension. So the files anchored on the website are identified by parsing html content for anchor tags.

The links for the files are stored in a list. The files are then ‘downloaded’ or written chunk by chunk and stored in the location of the folder and given their respective names.

We also have to identify the extensions and names of the files which are extracted from the links in the anchor tags and stored in another list. Then directories are formed according to the extensions stored in the list. The programs then iterate through every file and move them to their respective folder.

The program checks if a pdf file exists in the files downloaded if it does then the pdf parsing function is called else the program informs the user.

The links obtained are also stored in a file called ‘ Links.txt ’. The program prompts the user to name the folder in which all these files are stored.

## 5.2 PDF Parser

The text needed to be extracted from the pdf to check if the phrase given by the user appears in it. So every PDF downloaded had to be parsed successfully so that the user could receive accurate results.

The pdfminer module was used for this purpose and the text was extracted from the pdf and compared with the phrase but we also had to find if the pdf's are highly relevant. Count operation is used to solve this problem and files are then sorted depending on the number of counts for the phrase in descending order.

This Process is very time consuming and processing heavy task. So to overcome this issue Thread Pooling was used. Thread Pool was used to extract text from multiple pdf's and process them at the same time which helped to reduce the processing time by 78%. The names of the files are then displayed on the screen.

## 5.3 Related Phrase Searching

This task was supposed to provide accurate information while not being heavy on the processor so the only way to find relevant words was to browse the internet. The module Googlesearch was imported for the purpose of searching the phrase on the internet.

Wikipedia provided very relevant information about the phrase. The wikipedia page is then parsed for relevant words and displayed on the screen.

# 6. Implementation

## 6.1 Modules Used

### WebScraping(Main).py

```
import requests
from bs4 import BeautifulSoup
from os.path import split
import os
from urllib.parse import urlparse
import shutil
import PDF_Parser
import RelatedPhrase
```

### PDF_Parser.py

```
import io
import os
import threading
import queue
from multiprocessing.pool import ThreadPool
from pdfminer.pdfinterp import PDFResourceManager, PDFPageInterpreter
from pdfminer.converter import TextConverter
from pdfminer.layout import LAParams
from pdfminer.pdfpage import PDFPage
```

**RelatedPhrase.py**

```
from googlesearch import search
import requests
from bs4 import BeautifulSoup
import PDF_Parser
import urllib.request
```

# 6.2 Requirements

- Processors: Intel Atom® processor or Intel® Core™ i3 processor
- Disk space: 2 GB
- Operating systems: Windows* 7 or later, macOS, and Linux
- Python* version : 3.7.6
- Internet Access

# 7. Project Result

This project resulted in a program that can help students collect study materials and arrange them easily without any tedious work. This project helped me improve my problem solving skills and acquired a deeper understanding of Threading, Web Scraping and computation and in python programming.

Few alternate ways to parse pdf were tried which take less time but they are not reliable as more than often they are unable to successfully extract the text. The Speed at which the file is downloaded or written depends on the size of the particular file and lastly for finding similar words Wikipedia always gives best results as they have anchored links for related words.

Versions were also made using multiprocess.pool and multiprocess.process to speed up the parsing of PDF files as it is the only time consuming Process in the Program but due its parallel processing capabilities thread pooling proved most efficient.

This method is most efficient and very reliable due the combination of thread pool and use of a queue.

## 7.1 How To Use

1. Enter the address of the website when prompted or the url of the file you want to download.
2. Enter the Phrase You want to Search.
3. Then the Process will start and if the url is invalid or your internet connection is down, the program will inform you as so.
4. The Program will Display the http status response code.
5. The Program will Display which files are found while searching for files.
6. The program will ask you to name the folder in which the data will be stored.
7. The program will then check if the PDF's are available are not
8. If Yes , then the program will parse through the pdf's and display the top 5 relevant pdf's .
9. The program will then display similar phrases to the phrase entered by the user.
10. When the process is completed the downloaded Data will be stored in the location where the program file.

# 10. References


1. https://stackoverflow.com
2. https://www.youtube.com
3. http://www.geeksforgeeks.org
4. https://www.quora.com
5. https://www.google.com
6. https://www.wikipedia.org
7. https://docs.python.org/3/
8. https://www.tutorialspoint.com
9. https://www.w3schools.com