Assignment 7

1. Crawl.py:

```
import requests
from bs4 import BeautifulSoup
from urllib.parse import urljoin, urlparse

def normalize_url(url):
    """Normalizes the URL by removing trailing slashes and converting to lowercase."""
    parsed_url = urlparse(url)
    normalized_url = f"{parsed_url.hostname}{parsed_url.path}".rstrip('/')
    return normalized_url

def get_urls_from_html(html_body, base_url):
    """Extracts all URLs from the HTML body."""
    urls = []
    soup = BeautifulSoup(html_body, 'html.parser')
    for link_element in soup.find_all('a'):
        href = link_element.get('href')
        if href:
            if href.startswith('/'):
                # Relative URL, convert to absolute
                url = urljoin(base_url, href)
                urls.append(url)
            else:
                # Absolute URL
                urls.append(href)
    return urls

def crawl_page(base_url, current_url, pages):
    """Crawls a web page and retrieves all internal links."""
    parsed_base_url = urlparse(base_url)
    parsed_current_url = urlparse(current_url)

    if parsed_base_url.hostname != parsed_current_url.hostname:
        return pages

    normalized_current_url = normalize_url(current_url)
    if normalized_current_url in pages:
        pages[normalized_current_url] += 1
        return pages

    pages[normalized_current_url] = 1
    print(f"Actively crawling: {current_url}")

    try:
        response = requests.get(current_url)
        if response.status_code > 399:
            print(f"Error in fetching with status code: {response.status_code} on page: {current_url}")
            return pages

        if 'text/html' not in response.headers.get('Content-Type', ''):
            print(f"Non-HTML response, content type: {response.headers.get('Content-Type')} on page: {current_url}")
            return pages

        html_body = response.text
        next_urls = get_urls_from_html(html_body, base_url)

        for next_url in next_urls:
            pages = crawl_page(base_url, next_url, pages)
    except requests.RequestException as e:
        print(f"Error fetching from {current_url}: {e}")

    return pages
```

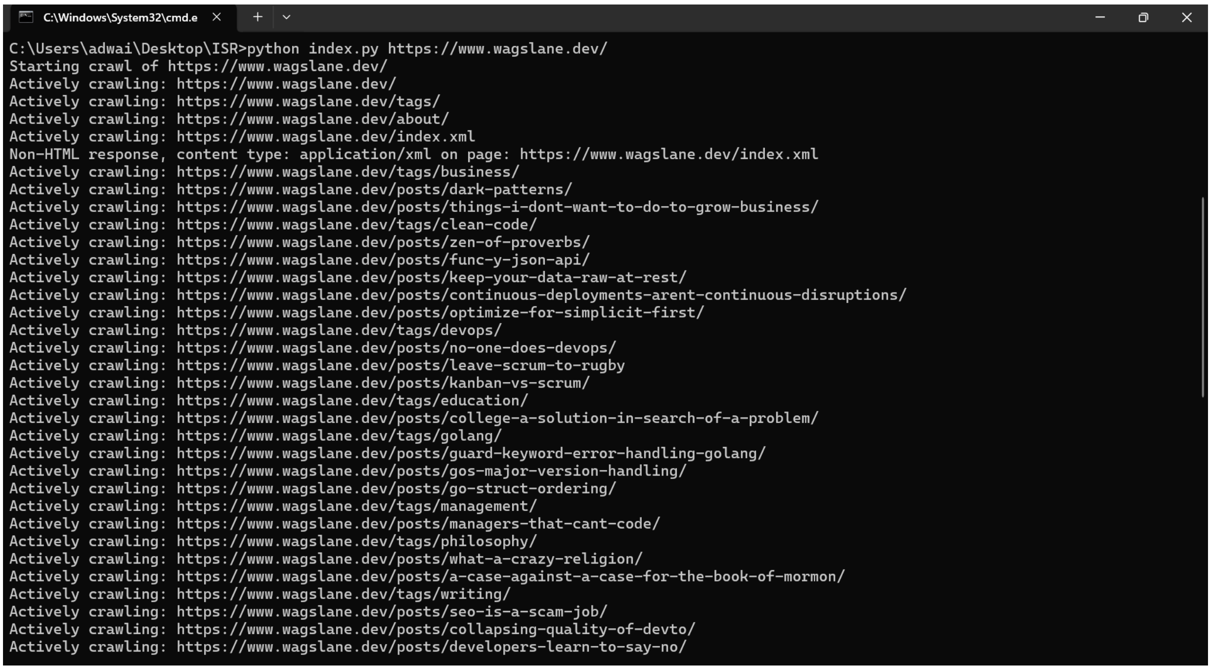
2. Index.py

```
import sys
from crawl import crawl_page
from report import print_report

def main():
    if len(sys.argv) < 2:
        print("No website provided")
        sys.exit(1)
    if len(sys.argv) > 2:
        print("Too many arguments!")
        sys.exit(1)

    base_url = sys.argv[1]
    print(f"Starting crawl of {base_url}")

    # Start the crawling process
    pages = crawl_page(base_url, base_url, {})

    # Print the report
    print_report(pages)

if __name__ == "__main__":
    main()
```

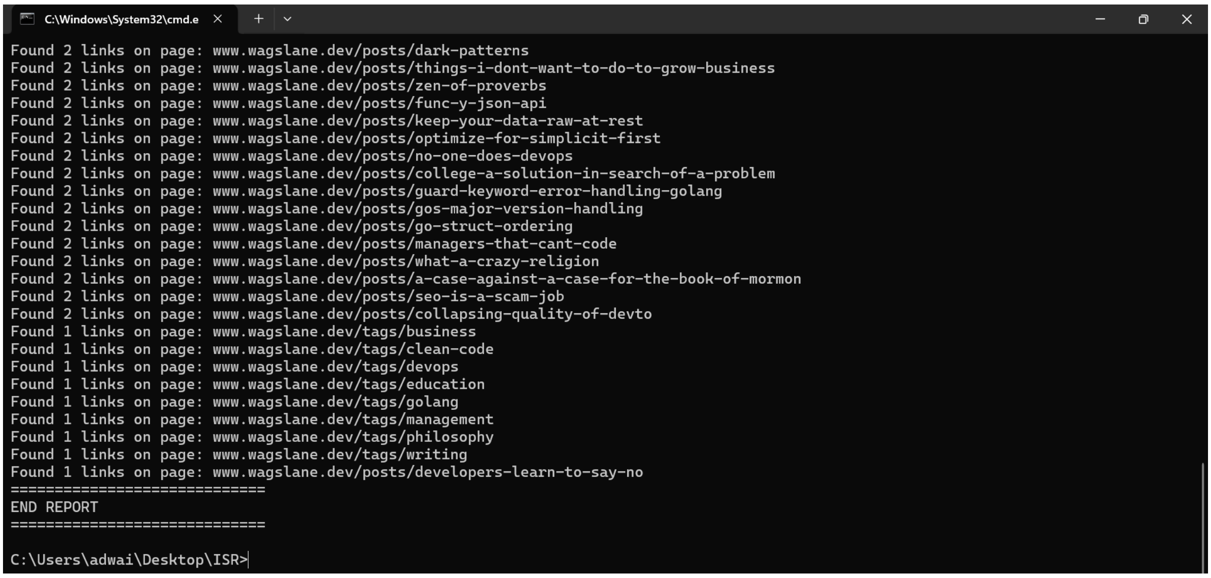
3. report.py

```
def print_report(pages):
    """Prints a report of all crawled pages and the number of links found."""
    print("==============================")
    print("REPORT")
    print("==============================")
    sorted_pages = sort_pages(pages)

    for url, hits in sorted_pages:
        print(f"Found {hits} links on page: {url}")

    print("==============================")
    print("END REPORT")
    print("==============================")

def sort_pages(pages):
    """Sorts pages based on the number of hits."""
    return sorted(pages.items(), key=lambda item: item[1], reverse=True)
```

OUTPUT:



```
C:\Users\adwai\Desktop\ISR>python index.py https://www.wagslane.dev/
Starting crawl of https://www.wagslane.dev/
Actively crawling: https://www.wagslane.dev/
Actively crawling: https://www.wagslane.dev/tags/
Actively crawling: https://www.wagslane.dev/about/
Actively crawling: https://www.wagslane.dev/index.xml
Non-HTML response, content type: application/xml on page: https://www.wagslane.dev/index.xml
Actively crawling: https://www.wagslane.dev/tags/business/
Actively crawling: https://www.wagslane.dev/posts/dark-patterns/
Actively crawling: https://www.wagslane.dev/posts/things-i-dont-want-to-do-to-grow-business/
Actively crawling: https://www.wagslane.dev/tags/clean-code/
Actively crawling: https://www.wagslane.dev/posts/zen-of-proverbs/
Actively crawling: https://www.wagslane.dev/posts/func-y-json-api/
Actively crawling: https://www.wagslane.dev/posts/keep-your-data-raw-at-rest/
Actively crawling: https://www.wagslane.dev/posts/continuous-deployments-arent-continuous-disruptions/
Actively crawling: https://www.wagslane.dev/posts/optimize-for-simplicit-first/
Actively crawling: https://www.wagslane.dev/tags/devops/
Actively crawling: https://www.wagslane.dev/posts/no-one-does-devops/
Actively crawling: https://www.wagslane.dev/posts/leave-scrum-to-rugby
Actively crawling: https://www.wagslane.dev/posts/kanban-vs-scrum/
Actively crawling: https://www.wagslane.dev/tags/education/
Actively crawling: https://www.wagslane.dev/posts/college-a-solution-in-search-of-a-problem/
Actively crawling: https://www.wagslane.dev/tags/golang/
Actively crawling: https://www.wagslane.dev/posts/guard-keyword-error-handling-golang/
Actively crawling: https://www.wagslane.dev/posts/gos-major-version-handling/
Actively crawling: https://www.wagslane.dev/posts/go-struct-ordering/
Actively crawling: https://www.wagslane.dev/tags/management/
Actively crawling: https://www.wagslane.dev/posts/managers-that-cant-code/
Actively crawling: https://www.wagslane.dev/tags/philosophy/
Actively crawling: https://www.wagslane.dev/posts/what-a-crazy-religion/
Actively crawling: https://www.wagslane.dev/posts/a-case-against-a-case-for-the-book-of-mormon/
Actively crawling: https://www.wagslane.dev/tags/writing/
Actively crawling: https://www.wagslane.dev/posts/seo-is-a-scam-job/
Actively crawling: https://www.wagslane.dev/posts/collapsing-quality-of-devto/
Actively crawling: https://www.wagslane.dev/posts/developers-learn-to-say-no/
```



```
Found 2 links on page: www.wagslane.dev/posts/dark-patterns
Found 2 links on page: www.wagslane.dev/posts/things-i-dont-want-to-do-to-grow-business
Found 2 links on page: www.wagslane.dev/posts/zen-of-proverbs
Found 2 links on page: www.wagslane.dev/posts/func-y-json-api
Found 2 links on page: www.wagslane.dev/posts/keep-your-data-raw-at-rest
Found 2 links on page: www.wagslane.dev/posts/optimize-for-simplicit-first
Found 2 links on page: www.wagslane.dev/posts/no-one-does-devops
Found 2 links on page: www.wagslane.dev/posts/college-a-solution-in-search-of-a-problem
Found 2 links on page: www.wagslane.dev/posts/guard-keyword-error-handling-golang
Found 2 links on page: www.wagslane.dev/posts/gos-major-version-handling
Found 2 links on page: www.wagslane.dev/posts/go-struct-ordering
Found 2 links on page: www.wagslane.dev/posts/managers-that-cant-code
Found 2 links on page: www.wagslane.dev/posts/what-a-crazy-religion
Found 2 links on page: www.wagslane.dev/posts/a-case-against-a-case-for-the-book-of-mormon
Found 2 links on page: www.wagslane.dev/posts/seo-is-a-scam-job
Found 2 links on page: www.wagslane.dev/posts/collapsing-quality-of-devto
Found 1 links on page: www.wagslane.dev/tags/business
Found 1 links on page: www.wagslane.dev/tags/clean-code
Found 1 links on page: www.wagslane.dev/tags/devops
Found 1 links on page: www.wagslane.dev/tags/education
Found 1 links on page: www.wagslane.dev/tags/golang
Found 1 links on page: www.wagslane.dev/tags/management
Found 1 links on page: www.wagslane.dev/tags/philosophy
Found 1 links on page: www.wagslane.dev/tags/writing
Found 1 links on page: www.wagslane.dev/posts/developers-learn-to-say-no
=============================
END REPORT
=============================

C:\Users\adwai\Desktop\ISR>
```