# Report on crawler.php and search.php

## Crawler.php

### Purpose and Functionality

The crawler.php script is designed to crawl a specified target URL (in this case, "[https://en.wikipedia.org/wiki/Napoleon"](https://en.wikipedia.org/wiki/Napoleon%22)) up to a certain depth limit, fetch the HTML content of each page, and store it in a specified output directory. Additionally, the script respects the rules specified in the robots.txt file to avoid crawling disallowed paths.

### Key Components and Actions

1. **Fetching and Parsing Robots.txt:** The script initiates by extracting the base URL and constructing the robots.txt URL. It then fetches and parses the robots.txt file to identify disallowed links.
2. **HTML Content Retrieval:** The script iterates through the URLs to scrape, retrieves the HTML content using cURL, and checks for errors or invalid content.
3. **HTML Parsing and Filtering:** The HTML content is parsed using DOMDocument, and the script extracts all anchor (<a>) tags. It filters and normalizes these URLs, discarding disallowed paths and already processed URLs.
4. **Saving HTML Content:** The final HTML content is saved to files in the specified output directory.

### Improvements and Considerations

1. **Error Handling:** The script lacks detailed error handling. Enhancements could be made to handle HTTP errors, malformed HTML, or other issues more robustly.
2. **Logging:** Introducing a logging mechanism would be beneficial for tracking the execution flow, errors, and other relevant information.
3. **Parallel Processing:** Implementing parallel processing for crawling multiple URLs simultaneously could improve efficiency.
4. **Configuration Management:** Externalizing configuration parameters, such as target URL and depth limit, would enhance flexibility.

## Search.php

### Purpose and Functionality

The search.php script performs a basic document search within the fetched HTML content using a specified search term. It calculates document scores based on the term frequency-inverse document frequency (TF-IDF) algorithm and presents the top results.

### Key Components and Actions

1. **Loading HTML Documents:** The script loads HTML documents from the specified output directory using DOMDocument.
2. **Text Extraction:** Text content is extracted from various HTML elements such as headings, paragraphs, spans, anchors, list items, table cells, labels, and buttons.
3. **Vectorization:** Both the search term and document content are vectorized using character frequency.
4. **Scoring:** Scores are calculated for each document based on the cosine similarity between the search term and document vectors.
5. **Result Presentation:** The top-scoring documents are presented as the search results, and the script highlights the search term occurrences in the output.

### Improvements and Considerations

1. **Algorithm Enhancement:** Consider using more advanced natural language processing (NLP) techniques for better search relevance.
2. **Scalability:** The script may need optimization for scalability with a larger number of documents.
3. **User Interaction:** Enhancements can be made for user interaction, such as presenting clickable links to the actual documents.
4. **Code Modularity:** Splitting the code into functions or classes for better modularity and maintainability could be considered.
5. **Search Result Format:** Depending on the use case, presenting search results in a more user-friendly format (e.g., a web page) could be beneficial.

## Overall Recommendations

1. **Documentation:** Include comments and documentation to improve code readability and assist future developers in understanding the code.
2. **Testing:** Implement thorough testing, especially for edge cases and error scenarios.
3. **Security:** Ensure that the script follows best practices for web scraping and document loading to avoid potential security vulnerabilities.
4. **Continuous Improvement:** Regularly update and enhance the scripts based on feedback, changing requirements, or advancements in technologies.
5. **User-Friendly Output:** Consider providing a more user-friendly output, especially for search results, to enhance the overall user experience.