**Multi-Threaded Web Crawler**

**Project Description**:

The Multi-Threaded Web Crawler is a Java-based application designed to efficiently crawl web pages starting from a given URL. It extracts links, processes them recursively, and handles multiple crawling tasks concurrently using a thread pool. This implementation ensures faster crawling, error handling, and the ability to resume interrupted sessions via URL persistence.

**Input**:

Starting URL: A user-defined URL to initiate the crawling process (e.g., http://books.toscrape.com).

Thread Pool Size: The number of concurrent threads.

Maximum Crawl Depth: The maximum levels of recursion for links.

**Output**:

Visited URLs: A list of URLs successfully crawled during execution.

Failed URLs: URLs that could not be processed due to errors, with one retry attempted.

Crawl Statistics:

Total URLs visited.

Total time taken for crawling.

URL Persistence: Visited URLs saved in visited\_urls.ser for resuming future crawls.

**Key Features**:

Multi-Threading: Concurrently handles multiple URL fetches using Java’s ExecutorService.

Error Handling: Retries failed URLs and logs permanent failures.

URL Persistence: Saves visited URLs to a file for resuming after interruptions.

Adjustable Thread Pool: Dynamically changes the number of threads during execution.

**Setup**:

Install JDK 8 or later.

Download and add the jsoup library to the project from https://jsoup.org/download.

**Run the Program**:

Execute the main method in Main.java using an IDE like IntelliJ IDEA.

This project showcases how to build a scalable, robust web crawler with multi-threading and persistence, making it a foundational tool for web scraping and data collection tasks.