Book Data Organizer and Metadata Extractor from Z-Library

https://z-lib.is/

Objective

Develop a Python script or application to scrape book data and cover images from Z-Library, categorize them based on book categories, store them in an organized file system, and create JSON files with detailed information about each book. The project will explore both serial and multithreaded programming approaches.

Project Description

Inputs

1. **Website URL**: The primary input will be a URL to the Z-Library book site.(https://z-lib.is/)
2. **Category and Book Data**: Identify and utilize data about book categories and related information present on the website.

Outputs

1. **Folder Structure**: Create a main directory with subdirectories for each book category. Within each category's folder, store book cover images and JSON files containing book metadata.
2. **Stored Book Images**: Download and store book cover images in their respective category folders.
3. **JSON Files**: Generate JSON files containing detailed information for each book, including title, author, ISBN, category, image URL, and other relevant metadata.
4. **Summary Report**: A chart summarizing the number of books downloaded per category.

Phases

1. **Phase 1: Serial Implementation**
    - Implement web scraping, categorization, file storage, and JSON file generation using a serial approach.
    - Document the performance in terms of execution time and resource utilization.
2. **Phase 2: Multithreaded Implementation**
    - Modify the script to perform the same tasks using multithreading.
    - Analyze and compare the performance with the serial approach.

GitHub Repository

- Actively contribute to and document the project on GitHub, with regular commits.

README File

- Include detailed instructions on setup, operation, and functionalities.

Project Report

1. Start with a list of all team members, including names and roles.
2. Outline the project's goals, architecture, strategy for web scraping, file organization, JSON file creation, and the transition from serial to multithreaded programming.
3. Compare the serial and multithreaded implementations in terms of efficiency and resource utilization.
4. Discuss the challenges faced, solutions implemented, and the overall learning experience.