Vector Graphic Organizer and Metadata Extractor from Pixabay

https://pixabay.com/vectors/

Objective

Develop a Python script or application to scrape vector graphics and their metadata from Pixabay, categorize them based on vector graphic categories, store them in an organized file system, and create a CSV file in each category folder containing detailed metadata of the vector graphics. This project will explore both serial and multithreaded programming approaches.

Project Description

Inputs

1. **Website URL**: The primary input will be the URL to Pixabay, specifically the section hosting vector graphics (https://pixabay.com/vectors/).
2. **Category and Vector Graphic Data**: Identify and utilize data about vector graphic categories and their related metadata present on the website.

Outputs

1. **Folder Structure**: Create a main directory with subdirectories for each vector graphic category. Each category's folder will store vector graphics files and a CSV file.
2. **Stored Vector Graphics**: Download and store vector graphics in their respective category folders.
3. **CSV Files**: Generate a CSV file in each category folder containing detailed information for each vector graphic, including title, tags, creator name, vector file URL, and other relevant metadata.
4. **Summary Report**: A chart summarizing the number of vector graphics downloaded per category.

Phases

1. **Phase 1: Serial Implementation**
   - Implement web scraping, categorization, file storage, and CSV 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, CSV 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.