Photo Organizer, Categorizer, and Metadata Extractor

https://stocksnap.io/

Objective Develop a Python script or application that can scrape photographs and their metadata from https://stocksnap.io/, categorize them based on the photographer and the photo's theme, store them in an organized file system, and create a CSV file with detailed information about each photo. The project will explore both serial and multithreaded programming approaches.

Project Description

Inputs

1. **Website URL:** The primary input will be a URL to a photography website. (https://stocksnap.io/)
2. **Photographer and Category Data:** Identify and utilize data about photographers and photo categories present on the website.

Outputs

1. **Folder Structure:** Create a main directory with subdirectories for each photographer. Within each photographer's folder, create further subdirectories for different photo categories.
2. **Stored Photographs:** Download and store photographs in their respective category folders.
3. **CSV File:** Generate a CSV file containing detailed information for each photo, including title, photographer, tags, image URL, and other relevant metadata.
4. **Summary Report:** A chart summarizing the number of photographs downloaded per category and per photographer.

Phase 1: Serial Implementation

- Implement web scraping, categorization, file storage, and CSV generation using a serial approach.
- Document the performance in terms of execution time and resource utilization.

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 student numbers.
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.