Multimedia Content Management System with Database Integration

Objective

The goal of this project is for you to develop a multimedia content management system that utilizes two CSV datasets. One dataset contains information about images, including paths, titles, tags, descriptions, categories, and photographer details. The second dataset includes articles with content, keywords, titles, categories, and writer details, where writers may also be photographers. You are tasked with creating a database using a system of your choice (MySQL, MongoDB, MariaDB, etc.) to store this data efficiently. Additionally, you will build a website using Django or Flask to allow users to interact with this data through various views and functionalities. You will also implement a script to enhance missing metadata using PhotoTag.ai.

Project Description

Inputs

1. Image Data CSV: Contains image path, title, tags, description, category, and photographer's personal details including a unique code.
2. Article Data CSV: Includes article content, keywords, title, category, and writer details, also identified by a unique code.

Outputs

1. Database: An efficiently structured database storing all data from the CSV files with proper relations and integrity constraints.
2.Website:A Django or Flask-based website with multiple navigation options, displaying images and articles, and personalized content pages.
3. Enhanced Metadata Script:A script to fill in missing image metadata using PhotoTag.ai.

Phase 1: Database Design and Implementation

Model the Data: Design a relational or document-based schema based on the dataset specifics.
Implement the Schema: Set up the database and implement the schema using SQL or NoSQL commands.
Data Import: Develop scripts to import data from CSV files into the database, ensuring data integrity and proper linkage between images, articles, and authorship.

Phase 2: Website Development

Framework Choice: Develop the website using Django or Flask, depending on your familiarity and project requirements.
Front-End Design: Create a responsive web interface with options for Homepage, Writer/Photographer, Images, and Articles.
Back-End Integration: Develop backend functionalities to fetch data from the database and serve it through the website.
Interactive Elements: Implement interactive elements such as category filters for images and articles and detailed pages for each item and author.

Navigation Features:

Homepage: Links back to the main page.
Images: Dropdown menu categorizes images. Clicking a category shows all images under it, with an 'All' option for all images.
Articles: Similar structure to Images for browsing articles by category.
Writer/Photographer: Lists all photographers and writers. Clicking a name shows all related images and articles.

Phase 3: Metadata Enhancement Script

Script Development: Write a Python script to check for missing titles, tags, or descriptions in the image data.
Integration with PhotoTag.ai: Use PhotoTag.ai's API to fetch enhanced metadata for images lacking these details.
Data Update: Update the database with the enhanced metadata from PhotoTag.ai.

GitHub Repository

Regular Updates: Maintain a GitHub repository for version control and project tracking with structured commits and documentation.
Documentation: Include detailed README files describing the setup, operation, and functionalities of the database and website.