**Enhanced Photo Gallery System**

A feature-rich photo gallery application that combines Python and C++ for efficient image management and processing. This hybrid application provides a modern GUI interface with powerful backend capabilities for organizing, editing, and searching your photo collection.

**Features**

* **Modern GUI Interface**: User-friendly interface built with PySide6 (Qt for Python)
* **Efficient C++ Backend**: Core data structures and algorithms implemented in C++ for performance
* **Image Management**:
  + Import photos with automatic metadata extraction
  + Organize photos by date, location, and custom tags
  + Search and filter functionality
  + Batch processing operations
* **Image Editing**:
  + Rotate, crop, and resize images
  + Adjust brightness and contrast
  + Apply effects (blur, sharpen, grayscale, sepia, invert)
* **Advanced Data Structures**:
  + AVL Tree for balanced search and retrieval
  + Trie for efficient tag/prefix searching
  + Priority Queues for quick access to recent/popular photos
  + HashMap for location-based photo lookup
* **Additional Features**:
  + Slideshow functionality
  + Metadata viewing and editing
  + View count tracking
  + Sorting by various criteria (date, size, popularity)

**Technical Details**

**Python Components**

* Frontend GUI using PySide6 (Qt for Python)
* Image processing using PIL (Python Imaging Library)
* Metadata extraction using exifread
* Bridge to C++ backend via subprocess calls

**C++ Components**

* SQLite database for persistent storage
* Custom data structures:
  + AVL Tree (balanced binary search tree)
  + Trie (prefix searching)
  + Priority Queue (max heap)
  + HashMap (location indexing)
  + LinkedList (sequential operations)
* Algorithms:
  + QuickSort for efficient sorting
  + Binary Search for date range queries
  + KMP String Matching for description searches

**Requirements**

* Python 3.6+
* PySide6
* PIL (Pillow)
* exifread
* C++ compiler (with C++11 support)
* SQLite
* nlohmann/json (C++ JSON library)

**Installation**

1. Clone the repository:
2. git clone https://github.com/yourusername/photo-gallery-system.git
3. cd photo-gallery-system
4. Install Python dependencies:
5. pip install -r requirements.txt
6. Compile the C++ components:
7. g++ -std=c++11 -o photo\_gallery photo\_gallery\_cli.cpp -lsqlite3
8. Run the application:
9. python photo\_gallery\_app.py

**Usage**

**Adding Photos**

1. Click "Add Photos" in the toolbar or select File → Add Photos
2. Select one or more image files
3. Edit metadata if needed (location, date, tags, description)
4. Click Save

**Viewing and Editing Photos**

1. Click on a thumbnail to view details
2. Use "Edit Metadata" to update photo information
3. Use "Edit Image" to modify the image (rotate, crop, adjust, etc.)

**Searching Photos**

1. Select a search type from the dropdown (Location, Tag, Date Range, Description)
2. Enter your search term
3. Click Search or press Enter

**Slideshow**

1. Select Edit → Slideshow from the menu
2. Use Previous/Next buttons to navigate through photos

**Batch Processing**

1. Select Edit → Batch Processing from the menu
2. Select photos and operations to perform
3. Click Process

**Project Structure**

* photo\_gallery\_app.py: Main Python application
* photo\_gallery\_cli.cpp: C++ backend implementation
* images/: Directory for stored photos
* photo\_gallery.db: SQLite database file (created on first run)

**Contributing**

Contributions are welcome! Please feel free to submit a Pull Request.

**Acknowledgments**

* PySide6/Qt for the GUI framework
* SQLite for the database engine
* nlohmann/json for C++ JSON support