Bookcase Database

("bookcase.db", "bc.db")

Programming Languages

- **JavaScript**
  - Use for building the user interface of website.
  - (handling image analysis)
- **Python**
  - server-side development & back-end processing.
  - interacting with database
  - (handling image analysis)
- **SQLite**
  - lightweight, efficient, use to store book information
- **HTML/CSS**
  - user interface

Frameworks/Tools

- **Flask**
  - handling HTTP requests, rendering templates, and managing back-end logic
- **SQLite**
- **JS Libraries -** use for reading book spine text from images
  - **OCR**
  - **TensorFlow.JS**
- **Cloud Vision API**
  - enhance image analysis
- **ChatGPT API**
  - enhance image analysis

Project Steps

1. **Set Up Flask App:**
   - Create a Flask application with routes for uploading images and processing requests.
2. **HTML/CSS:**
   - Develop a clean and user-friendly interface for image upload and display.
3. **Image Processing:**
   - Integrate JavaScript libraries or APIs for image analysis to read text from the book spines.
4. **Database Integration:**
   - Use SQLite to store information about books, including title, author, and ISBN.
5. **Constant Book Identification:**
   - Implement a mechanism to identify a constant book in the image. This could involve matching specific characteristics like font or unique features.
6. **Image Correction:**
   - Use the identified constant book to correct the lighting and perspective of the entire image, enhancing accuracy in text recognition.
7. **Book Information Retrieval:**
   - Leverage APIs or web scraping to fetch book details, including ISBN, based on the recognized text.
8. **Export to Spreadsheet:**
   - Develop functionality to export the gathered book information to a downloadable spreadsheet.
9. **Testing:**
   - Rigorously test application with various images to ensure accurate recognition and data extraction.
10. **Deployment:**
    - Deploy Flask application to a web server, making it accessible to users.