Manage Library Books - Documentation

1. Setup and Run on a New Local Machine
Prerequisites: - Docker Desktop installed and running - Composer installed (to fetch CodeIgniter dependencies if not already included) - Git (optional, for cloning the repository)

Steps:

1. Clone or download the project: git clone https://github.com//Manage-Library-Books.git cd Manage-Library-Books
2. Start the containers with Docker Compose: docker compose up -d --build
   This will start: - mlb_app -> PHP 8.2 + Apache (CodeIgniter app) - mlb_db -> MySQL 8 database - mlb_pma -> phpMyAdmin for DB management
3. Configure CodeIgniter environment:
   cd app
   cp env .env
   php spark key:generate
   Then edit .env and set DB config:
   database.default.hostname = db
   database.default.database = library
   database.default.username = app
   database.default.password = app
4. Run database migrations:
   php spark migrate
   php spark db:seed BookSeeder
5. Access the application:
   - App: http://localhost:8080
   - phpMyAdmin: http://localhost:9090 (user: app, pass: app)

2. Development and Design Decisions

Framework Choice (CodeIgniter 4):
Chosen for lightweight MVC structure and simplicity in CRUD development.

- Containerization (Docker + Compose):
Ensures consistent environment setup across machines with one command.

- Database Schema: A single 'books' table stores title, author, genre, year, and optional cover image. - Validation Rules:

- Title, Author, Year are required - Genre, Image are optional - Year must be numeric and realistic

- Image Handling: Uploaded images saved in writable/uploads. If no image is provided, a placeholder ("No image") is shown.

- UI & Styling: Bootstrap 5 used for responsive and clean design. - Flash Messages: User feedback provided after create/update/delete actions.

- Security Decisions:
- Input validation with CI4 rules
- Images renamed with random names to avoid conflicts
- Models used to prevent SQL injection risks

- Scalability:
The project is structured to allow adding pagination, search, or categories in the future without major refactoring.