Library Management System (Django)

A complete Library Management System built with Django. It includes custom authentication, book CRUD, filtering/search, favorites, admin-only bulk deletion, categories management, sample data population, and tests.

Features

- Add/Edit/Delete Books (admin only)
- List Books with rich filters:
  - Title, Author, Category
  - Price range, Publish date range
  - Availability status
- Delete all filtered books (admin only)
- Search by title and author (separate fields)
- User accounts: signup, login, logout (custom user with roles: member, admin)
- Favorite books per user (toggle)
- Categories: create/list/delete (admin only)
- Bootstrap 5 UI
- Management command to populate ~100 books with realistic data
- Tests for CRUD, search/filter, delete filtered, and auth

Tech

- Python 3.11+
- Django 4.x+ (works on 5.x)
- SQLite (development)
- Bootstrap 5

Quickstart

```
# Windows PowerShell
cd "E:\Repositories\Quera Django Bootcamp\personal_project\library_management_system"
..\venv\Scripts\python.exe -m pip install --upgrade pip
..\venv\Scripts\python.exe -m pip install "Django>=4,<6"
..\venv\Scripts\python.exe manage.py makemigrations
..\venv\Scripts\python.exe manage.py migrate
..\venv\Scripts\python.exe manage.py createsuperuser
..\venv\Scripts\python.exe manage.py populate_db
..\venv\Scripts\python.exe manage.py runserver
```

Open `http://127.0.0.1:8000/`.

Apps and Structure

- `users`:
  - Custom `User` model (`full_name`, `role`: member/admin)
  - Forms: `SignupForm`, `LoginForm`
  - Views: `signup_view`, `login_view`, `logout_view`
  - URLs: `/users/signup/`, `/users/login/`, `/users/logout/`
- `books`:
  - Models: `Publisher`, `Category` (self-referential), `Author`, `Book`, `FavoriteBook`
  - Views:
    - `book_list` (filtering/search, shows all results by default)
    - `book_add`, `book_edit`, `book_delete` (admin only)
    - `delete_filtered_books` (admin only)
    - `toggle_favorite` (auth users)
    - `favorites_list` (auth users)
    - `category_list`, `category_add`, `category_delete` (admin only)
  - Templates include Bootstrap-based navbar and pages for list, form, delete, favorites, and categories
  - Management command: `populate_db` creates ~100 books and related data

Filtering

- GET parameters on `/books/`:
  - `title`, `author`, `category` (id), `availability`
  - `min_price`, `max_price` (numeric)
  - `start_date`, `end_date` (YYYY-MM-DD)
- The page shows a “Delete Filtered” button (admin) that POSTs with the exact same filters.

Permissions

- Members:
  - Browse books, filter/search, favorite/unfavorite, view favorites
- Admins:
  - All the above + create/edit/delete books, delete filtered, manage categories

Tests

Run tests:

```
..\venv\Scripts\python.exe manage.py test
```

Notes

- `AUTH_USER_MODEL` is set to `users.User` in `settings.py`.
- All FKs to user use `settings.AUTH_USER_MODEL`.
- Favorites are unique per user-book.
- Pagination was removed per request to simplify browsing large datasets.

ERD

See `Library management system ERD.pdf`.

Git

- `.gitignore` is configured for Python/Django/venv/SQLite.