Steps to follow for building and run this Dockerized project:

1. Install Docker Desktop:
    - Download and install Docker Desktop for Windows from the official Docker website: Get Started | Docker

2. You have already pull the github project

3. Navigate to *Task2\book_service*

4. Build the Docker Image:
    - Build the Docker image using the docker build command. Make sure to include the . at the end, which indicates the current directory:
      docker build -t yourusername/your-image-name .

    Replace "yourusername" and "your-image-name" with your desired Docker image name.

5. Run the Docker Container:
    - Once the image is built, you can run the Docker container:
      docker run -p 8080:80 yourusername/your-image-name

6. Access the Application:
    - Open a web browser and navigate to *http://localhost:8080* to access your application. You can get the fastapi auto generate docs
      http://localhost:8000/docs#/

Previously there you see I wrote "yourusername" and "your-image-name," I'm referring to placeholders where you should provide your own unique names. Here's what you can do:

1. Choose a Docker Image Name
   Decide on a name for your Docker image. This can be anything you like, but it's common to use a combination of your project name, organization, or your own username. For example:
   mybookapp

2. Tag Your Image
   Optionally, you can add a tag to your Docker image to indicate a version. If you don't specify a tag, Docker will use "latest" by default. For example:
   mybookapp:latest

3. Build and Run

4. Now, when you build your Docker image, use the name you decided on:
   docker build -t mybookapp .

And when you run the Docker container, use the same name:
d*ocker run -p 8080:80 mybookapp*

This is just an example, and you should replace "mybookapp" with your actual chosen name throughout the process. The -t flag in the docker build command is used to specify the name and optionally a tag for the image. The same name is then used in the docker run command.

Details about api routes -

Admin Login

- Description: Login for admin.
- Endpoint: POST /api/admin/login/
- Parameters:
    - user_login: AdminLogin model (username and password)
- Response:
    - Success (200 OK):
        - Returns an access token for the admin.

Register Client

- Description: Register a client.
- Endpoint: POST /api/register/
- Parameters:
    - client_data: ClientCreate model (full name, email, password)
- Response:
    - Success (201 Created):
        - Returns the newly created client details.

Login Client

- Description: Login for client.
- Endpoint: POST /api/login/
- Parameters:
    - client_data: ClientLogin model (email and password)
- Response:
    - Success (200 OK):

        - Returns an access token for the client.

Add Book

- Description: Add a book. (Accessible only to admin)
- Endpoint: POST /api/books/
- Parameters:
    - book_data: BookCreate model (book title and author)
- Response:
    - Success (201 Created):
        - Returns the details of the added book.

Edit Book

- Description: Edit a book. (Accessible only to admin)
- Endpoint: PUT /api/books/{book_id}
- Parameters:
    - book_id: Path parameter for the book to edit
    - book_data: BookUpdate model (new book title and author)
- Response:
    - Success (200 OK):
        - Returns the details of the edited book.

Retrieve List of Books

- Description: Retrieve list of books with filtering options. (Accessible only to admin)
- Endpoint: GET /api/books/
- Parameters:
    - title_start: Query parameter for filtering by the first letter of the book's title
    - author_id: Query parameter for filtering by author
- Response:
    - Success (200 OK):
        - Returns a list of books based on the filtering options.

Add Books by Author

- Description: Add multiple books by the same author. (Accessible only to admin)
- Endpoint: POST /api/authors/{author_id}/books/
- Parameters:
    - author_id: Path parameter for the author
    - book_titles: List of book titles to add
- Response:

        - Success (201 Created):
            - Returns details of the added books.

Add Author

- Description: Add author. (Accessible only to admin)
- Endpoint: POST /api/authors/
- Parameters:
    - author_data: AuthorCreate model (full name of the author)
- Response:
    - Success (201 Created):
        - Returns details of the added author.

Retrieve List of Authors

- Description: Retrieve list of authors with filtering options. (Accessible only to admin)
- Endpoint: GET /api/authors/
- Parameters:
    - full_name_start: Query parameter for filtering by the first letter of the author's full name
- Response:
    - Success (200 OK):
        - Returns a list of authors based on the filtering options.

Create Client

- Description: Create client. (Accessible only to admin)
- Endpoint: POST /api/clients/
- Parameters:
    - client_data: ClientCreate model (full name, email, password)
- Response:
    - Success (201 Created):
        - Returns details of the created client.

Retrieve List of Clients

- Description: Retrieve list of clients with filtering options. (Accessible only to admin)
- Endpoint: GET /api/clients/
- Parameters:
    - full_name_start: Query parameter for filtering by the first letter of the client's full name

- Response:
    - Success (200 OK):
        - Returns a list of clients based on the filtering options.

Retrieve Books Borrowed by Client

- Description: Retrieve list of books borrowed by the client. (Accessible only to client)
- Endpoint: GET /api/clients/books/
- Response:
    - Success (200 OK):
        - Returns a list of books borrowed by the client.

Link Client to Book

- Description: Link client to a book. (Accessible only to client)
- Endpoint: POST /api/clients/books/link/
- Parameters:
    - book_data: BorrowCreate model (book ID)
- Response:
    - Success (201 Created):
        - Returns details of the linked book.

Unlink Client from Book

- Description: Unlink client from a book. (Accessible only to client)
- Endpoint: POST /api/clients/books/unlink/
- Parameters:
    - request: UnlinkRequest model (borrow ID)
- Response:
    - Success (200 OK):
        - Indicates successful unlinking of the client from the book.