FastLib

MongoDB Database Setup

- Set up a MongoDB database named "library" to store information about the library's books.
- Design a collection named "books" to store book documents with attributes such as `title`, `author`, `publication_year`, `genre`, `availability_status`, and any additional fields you choose.

FastLib

Authentication Endpoints:

- Implement the `/register` endpoint to allow users to register an account. When a user registers, they should provide a unique username and password. You can store the user information in a separate collection named "users" in the MongoDB database.
- Implement the `/login` endpoint to verify the user's credentials. If the provided username and password match a registered user, generate a JSON Web Token (JWT) and return it to the user. The JWT will be used for authentication in subsequent requests.

FastLib

Book Management Endpoints:

- Implement the `/books` endpoint to retrieve a list of all books in the library. This endpoint should return a JSON response containing information about each book, such as its `title`, `author`, and `availability_status`. Additionally, implement the ability to add a new book to the library using the appropriate HTTP method.
- Implement the `/books/{book_id}` endpoint to retrieve details of a specific book by its `book_id`. This endpoint should return a JSON response containing all the available information about the book. Additionally, provide functionality to update the information of a book or remove it from the library.
- Implement the `/books/search` endpoint to enable users to search for books based on criteria such as `title`, `author`, or `genre`. This endpoint should return a JSON response containing a list of books that match the search criteria.
- Implement the `/books/{book_id}/borrow` endpoint to allow users to borrow a specific book. When a user makes a request to borrow a book, you need to check if the book is currently available (`availability_status`). If it is available, update the book's status to "borrowed" and associate the book with the user who borrowed it. If the book is already borrowed, return an appropriate error response.
- Implement the `/books/{book_id}/return` endpoint to enable users to return a borrowed book. When a user makes a request to return a book, check if the book is associated with the user who borrowed it. If the book is associated with the user, update the book's status to "available" and remove the association with the user. If the book is not associated with the user or it is already available, return an appropriate error response.
- Implement the `/books/genres` endpoint to retrieve a list of available book genres. This endpoint should return a JSON response containing a list of all the unique genres present in the library's collection of books.
- Implement the `/books/authors` endpoint to retrieve a list of all authors in the library. This endpoint should return a JSON response containing a list of all the unique authors present in the library's collection of books.
- Implement the `/books/{book_id}/reviews` endpoint to retrieve or add reviews for a specific book. This endpoint should provide functionality to retrieve existing reviews for the book or allow users to add their own reviews. You can choose the structure of the review document and define the necessary fields, such as `reviewer_name` and `comments`.

FastLib

Authentication and Authorization

- Secure the API endpoints using JWT authentication. When a user registers or logs in, generate a JWT containing the user's information and return it to the client. For subsequent requests, the client should include the JWT in the request headers for authentication.
- Implement authorization checks to ensure that users can only perform actions they are authorized to. For example, a user should only be able to modify their own reviews or borrow books. Verify the user's identity and authorization before allowing them to perform such actions.