**Backend Functional Requirements**

**Name: BookNest**

**User Authentication – done on 5-Nov-2024**

* **Registration:**
  + Users can create an account with email and password.
  + Users should receive a confirmation email (optional).
* **Login/Logout:**
  + Users can log in using their email and password.
  + Users can log out of the platform.
* **Profile Management:**
  + Users can view and edit their profiles (username, email, bio).
  + Option to upload a profile picture.

**Book Management**

* **Browse Books:**
  + Users can view a list of books with pagination.
  + Each book entry includes title, author, genre, and average rating.
* **Search Functionality:**
  + Users can search for books by title, author, or genre.
  + Filters for categories, ratings, and publication year.
* **Add Books:**
  + Users can submit new books, including title, author, genre, and a cover image.
  + Admin users can approve submitted books for public visibility.

**Review System**

* **Add Reviews:**
  + Users can write reviews for books they have read.
  + Reviews should include a rating (1-5 stars), a title, and a body.
* **Edit/Delete Reviews:**
  + Users can edit or delete their reviews.
* **View Reviews:**
  + Users can view reviews for each book.
  + Average rating displayed based on user ratings.

**Reading Lists**

* **Track Reading Progress:**
  + Users can categorize books into “Read”, “Currently Reading”, and “Want to Read”.
* **Bookshelf Feature:**
  + Users can create and manage custom bookshelves (e.g., favorite books, genres).

**Discussion Forum**

* **Book Clubs or Genre Discussions:**
  + Users can create or join discussion threads for specific books or genres.
* **Commenting System:**
  + Users can comment on reviews and engage in discussions.

**Design of the frontend:**

**Main Components**

* **Navbar: Include links to Home, Browse Books, My Profile, and a toggle for Dark Mode.**
* **Auth Components: Separate components for Login and Registration that will handle user authentication.**
* **Book Components:**
  + **BookList: Display all books with options to filter and search.**
  + **BookDetail: Show detailed information about a selected book and user reviews.**
  + **AddBook: Form for adding new books (accessible to admins).**
* **Review Components:**
  + **ReviewList: List reviews for each book.**
  + **AddReview: Form for users to submit a review.**
* **ReadingList Component: Manage reading progress and custom bookshelves.**
* **Forum Components: Create and join discussions.**

**3. User Authentication Flow**

* **Registration:**
  + **User fills out the registration form (email, password, username).**
  + **On submission, call the registration API to create a new user.**
  + **Send a verification email after successful registration.**
* **Login:**
  + **User enters email and password.**
  + **On submission, call the login API to authenticate the user.**
  + **Store the JWT token in local storage and update the application state.**
* **Logout:**
  + **Clear the JWT token from local storage and update the application state.**

**4. Book Management Flow**

* **Browse Books:**
  + **Fetch the list of books from the Django API on component mount.**
  + **Implement search and filter functionalities using local state.**
* **Add Book:**
  + **Admin can access the Add Book form to submit a new book.**
  + **Call the API to add the book and notify the admin of success/failure.**

**5. Review System Flow**

* **Add Review:**
  + **Allow users to submit reviews for books they have read.**
  + **Call the API to submit the review and update the state to show the new review.**
* **View Reviews:**
  + **Fetch and display reviews for each book on the Book Detail page.**

**6. Reading Lists Flow**

* **Manage Reading Lists:**
  + **Users can categorize books into "Read," "Currently Reading," and "Want to Read."**
  + **Call APIs to update the user’s reading list based on their selections.**

**7. Discussion Forum Flow**

* **Create/Join Discussions:**
  + **Allow users to create discussion threads for specific books or genres.**
  + **Fetch and display comments for each thread.**

**8. Dark Mode Implementation**

* **Use CSS classes to toggle between light and dark themes.**
* **Store the user's theme preference in local storage to maintain consistency.**

**9. Styling with Tailwind CSS**

* **Utilize Tailwind CSS for rapid and responsive design.**
* **Apply classes directly to your JSX elements for styling.**