**Functionality Guide**

**Index Page**

**Authentication Workflow:**

* **Login Process**:
  + Triggers when the "Login" button is clicked, displaying the login modal.
  + Upon form submission, gathers user input (username and password) and sends a POST request to the **/server/index.php/api/login** endpoint.
  + If successful (**resCode === 0**), redirects to the Home page (**/pages/home.html**); otherwise, displays an error message.
* **Registration Process**:
  + The registration form includes client-side validation:
    - Validates the email format.
    - Ensures the password is at least 8 characters long.
    - Checks password confirmation matches the initial password.
  + Upon validation success, sends a POST request to **/server/index.php/api/register**.
  + Upon successful registration, displays a success alert and redirects to the Home page.

**Home Page**

**Adding and Modifying Books:**

* **Add Book**:
  + Users fill out a form in a modal window with book details.
  + On submission, data is sent via POST to **/server/index.php/api/post/books**.
  + The book list is refreshed to include the new book entry.
* **Modify Book Details**:
  + Similar to adding a book, but pre-populates the modal form with the selected book's existing data.
  + On submission, updates are sent via PUT to **/server/index.php/api/put/books**.
  + The book list is refreshed to reflect the updates.

**Searching for Books:**

* Allows users to search by title, author, genre, or ISBN.
* Search results are fetched and displayed in the books table, with pagination support for navigation through results.

**Bookshelf Page**

* Displays a user-specific list of books added to their bookshelf.
* Each book card includes a "Remove" button:
  + Clicking this button sends a POST request to **/server/index.php/api/delete/userbook** to remove the book from the user's bookshelf.
  + The bookshelf display is refreshed to exclude the removed book.

**Book Detail Page**

* Fetches and displays detailed information about a specific book, including comments.
* **Add to Bookshelf**:
  + Allows users to add the displayed book to their personal bookshelf with a POST request to **/server/index.php/api/post/userbook**.
* **Commenting**:
  + Users can submit comments through a modal form.
  + Comments are sent via POST to **/server/index.php/api/post/bookcomment** and are dynamically added to the comment section.

**Dynamic Content Loading and User Interaction**

* The application heavily relies on the Fetch API for asynchronous data retrieval and submission, allowing for a responsive user experience without full page reloads.
* Bootstrap's modal components are utilized across the application for displaying forms and alerts, providing a consistent and modern UI/UX.
* Client-side JavaScript handles data binding, event listening, and dynamic content updates, ensuring the UI stays in sync with the backend state.

**Additional Technical Insights**

* **Form Data Handling**: Utilizes **FormData** objects and JSON serialization for preparing user input for HTTP requests.
* **Client-side Validation**: Enhances user experience by providing immediate feedback on input errors before submission to the server.
* **Error Handling**: Includes catch blocks for fetch operations to handle network or server errors gracefully.