Library Management

IIHT

Time To Complete: 2 hr

Contents

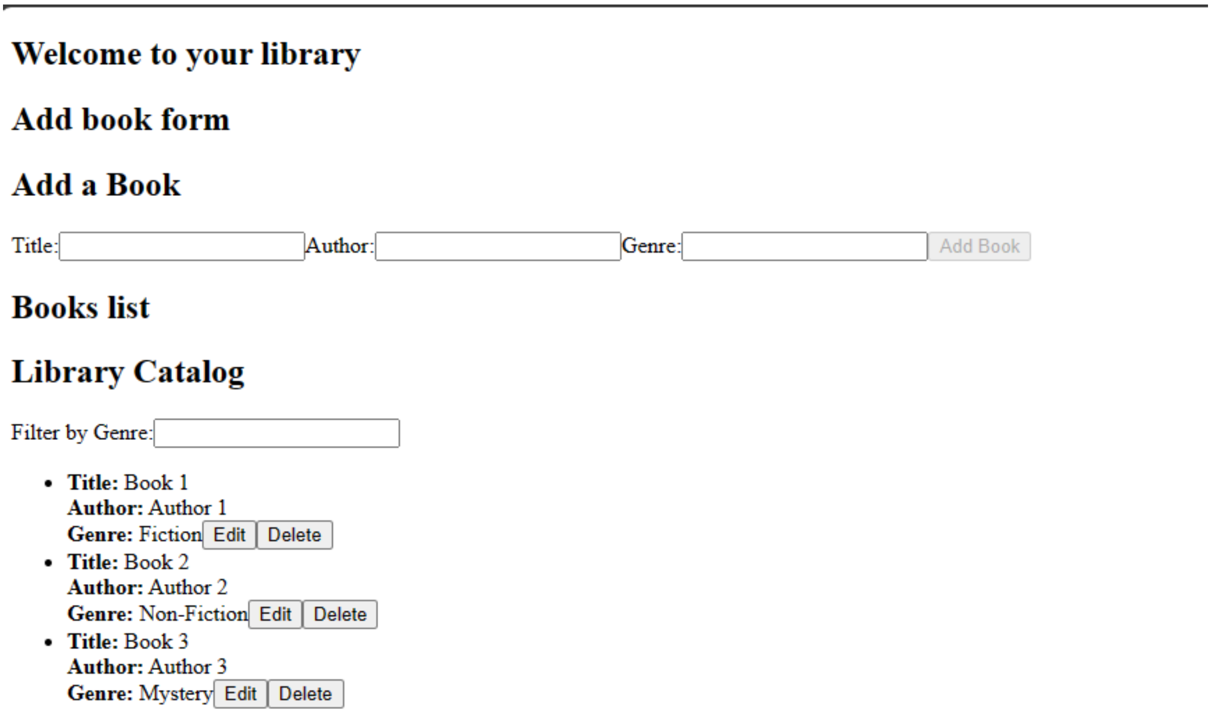
1 Problem Statement

"Library Management" is a Single Page Application (SPA) that empowers users to have a virtual library where they can search out for any book, get details of a book and also filter out books according to their genre.

2 Proposed Library Management Wireframe

UI needs improvisation and modification as per given use case and to make test cases passed.

2.1 Welcome Page



Welcome to your library

Add book form

Add a Book

Title: Author: Genre:

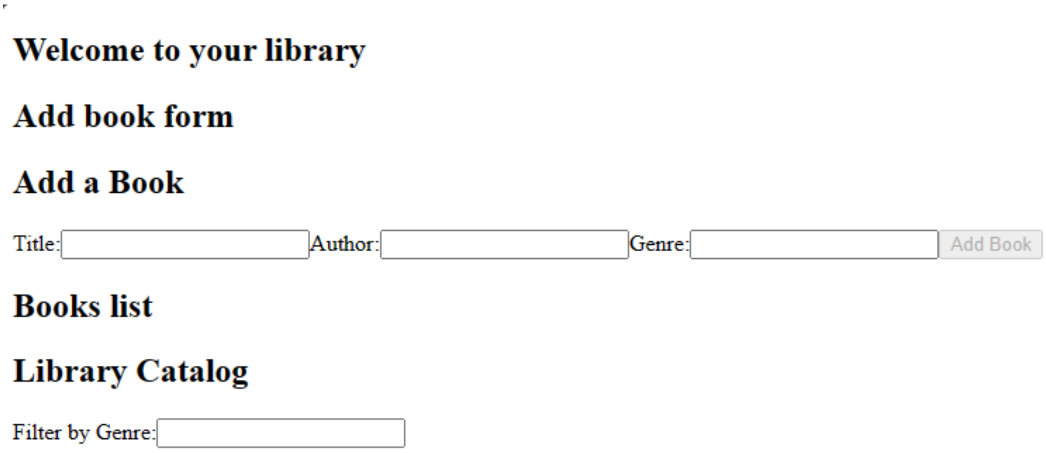
Books list

Library Catalog

Filter by Genre:

- **Title:** Book 1
  Author: Author 1
  Genre: Fiction
- **Title:** Book 2
  Author: Author 2
  Genre: Non-Fiction
- **Title:** Book 3
  Author: Author 3
  Genre: Mystery

Screen Shots



Welcome to your library

Add book form

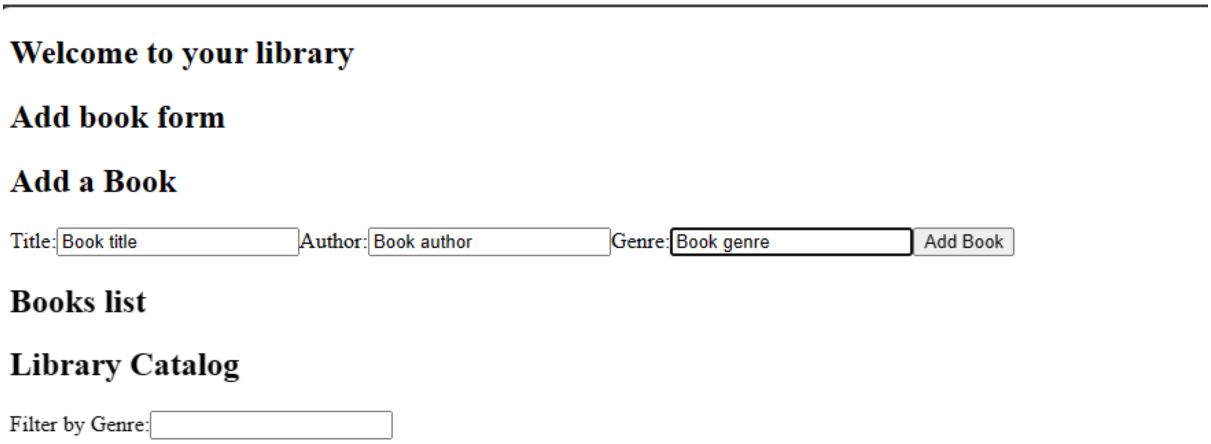
Add a Book

Title: Author: Genre:

Books list

Library Catalog

Filter by Genre:



Welcome to your library

Add book form

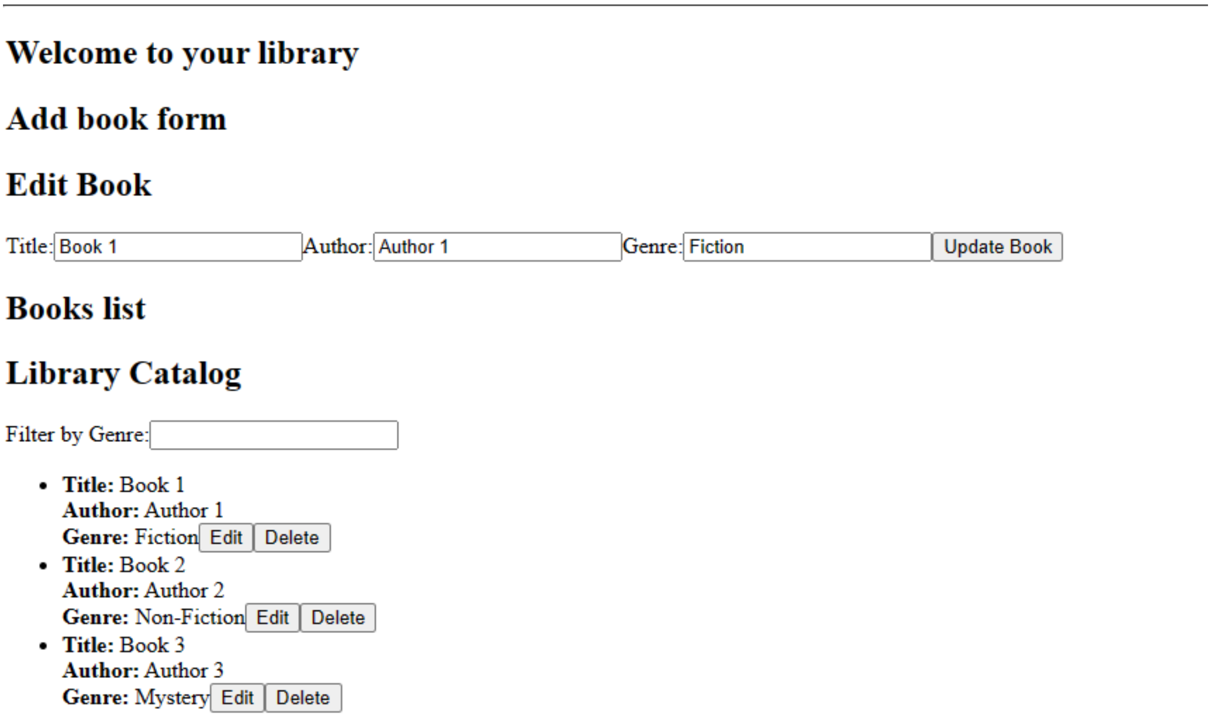
Add a Book

Title: Book title Author: Book author Genre: Book genre

Books list

Library Catalog

Filter by Genre:



Welcome to your library

Add book form

Edit Book

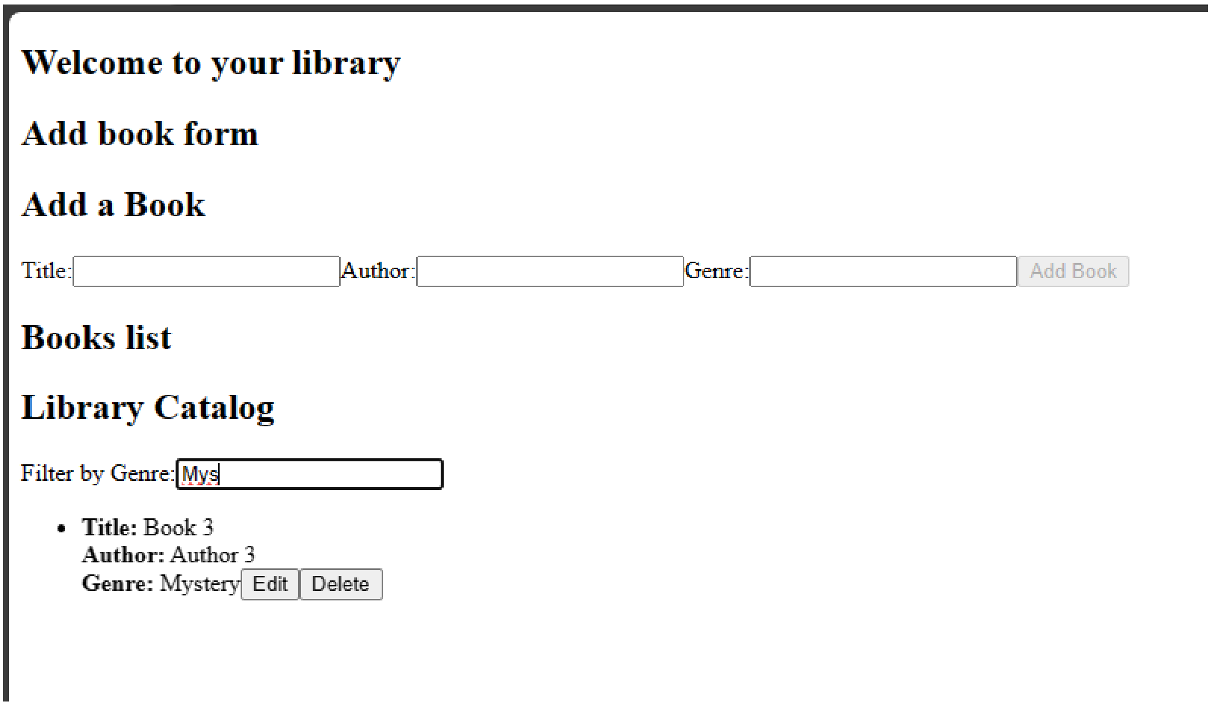
Title: Book 1 Author: Author 1 Genre: Fiction

Books list

Library Catalog

Filter by Genre:

- **Title:** Book 1
  Author: Author 1
  Genre: Fiction
- **Title:** Book 2
  Author: Author 2
  Genre: Non-Fiction
- **Title:** Book 3
  Author: Author 3
  Genre: Mystery



Welcome to your library

Add book form

Add a Book

Title: Author: Genre:

Books list

Library Catalog

Filter by Genre: Mys

- **Title:** Book 3
  Author: Author 3
  Genre: Mystery

3 BUSINESS-REQUIREMENT:

As an application developer, develop the Library Management (Single Page App) with below guidelines:

User Story #	User Story Name	User Story
US_01	Welcome Page	As a user I should be able to visit the welcome page as default page. Acceptance criteria: 1. Users should be able to see a form to add a book. 2. It must have Title, Author and Genre fields along with Add Book button. 3. Users should be able to see the list of all books. 4. Users should be able to filter out books by their genre.

4 VALIDATIONS

- All required fields must be fulfilled with valid data.
- In the starting “Add Book” button should be disabled.
- Only after validating fields, “Add Button” button should be enabled.
- Once we click on “Add Book”, a new book must be added in list.

5 CONSTRAINTS

- You should be able to press the “TAB” key and “SHIFT + TAB” to navigate from top field to bottom field and vice-versa.
- On clicking the “Add Book” button, a new book must be added with entered fields.
- “Add Book” button will be disabled until all validations are fulfilled.

6 Mandatory Assessment Guidelines

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to**
   Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.
3. **This editor Auto Saves the code.**
4. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.**
5. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**
6. **To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN.**
7. **This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.**
   Note: The application will not run in the local browser

8. **You can follow series of command to setup React environment once you are in your project-name folder:**
   a. **npm install -> Will install all dependencies -> takes 10 to 15 min**
   b. **npm run start -> To compile and deploy the project in browser. You can press <Ctrl> key while clicking on localhost:8080/8081 to open project in browser -> takes 2 to 3 min**
   c. **npm run json-start -> As we are using a json server to mimic our db.json file as a database. So, this command is useful to start a json server.**

   d. **npm run test -> to run all test cases. It is mandatory to run this command before submission of workspace -> takes 5 to 6 min**

9. **You may also run “npm run jest” while developing the solution to re-factor the code to pass the test-cases.**
10. **Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on “Submit Assessment” after you are done with code.**
11. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**