**EXPERIMENT 3.2**

### 🎯 Aim

To design and implement a Library Management User Interface (UI) using React.js that allows users to:

1. Search books by title or author dynamically.
2. Add new books to the collection.
3. Remove existing books from the collection.

This system helps manage books easily and provides a clean, interactive interface for users.

### 📚 Theory

The project is a single-page application (SPA) built with React.js, focusing on state management and dynamic rendering of UI elements.

#### 🔹 Key Concepts Used:

1. React Functional Components:  
   The entire application is structured as a functional component called LibraryManagement.
2. useState Hook:
   * books: Stores the list of book objects (with id, title, and author).
   * searchTerm: Stores the current search query.
   * newTitle and newAuthor: Hold input values for adding new books.
3. CRUD Operations:
   * Create: Adding a new book updates the books state dynamically.
   * Read: All books are displayed in a list.
   * Delete: Books can be removed by filtering them out of the state array.  
     (Update isn’t explicitly implemented here.)
4. Filtering (Search Functionality):  
   Uses JavaScript’s filter() and includes() to display only books that match the search term in title or author.
5. Dynamic Rendering:  
   React automatically re-renders the UI whenever the books state changes, ensuring instant updates.
6. Reusable and Scalable:
   * The UI is simple but scalable for a larger library database.
   * Could later integrate with a backend (Node.js, Express, or database).

