# **My Personal Library**

## **Overview**

You will build a **multi-page web application** called **My Personal Library**. The purpose of this project is to help you learn **DOM manipulation**, **form handling**, and **HTML/CSS layout** by doing. You have **2 full days** to complete this assignment. After that, we’ll review your work together and learn best practices.

## **Goal**

Create a personal library app that allows users to:

* View a list of books
* Add new books
* Edit or delete existing books
* Submit a contact form

## **Pages to Build**

1. **Home Page**
   1. Welcome message
   2. Brief description of the app
2. **Library Page**
   1. Displays a list of books with **CRUD** functionality:
      1. Each book has: id, Title, Author, Genre, Year Published
3. **Add Book Page**
   1. Form to add a new book
   2. Fields:
      1. Title (text)
      2. Author (text)
      3. Genre (dropdown)
      4. Year Published (number)
   3. Add form validation (e.g., required fields)
4. **Contact Page**
   1. Contact form
   2. Fields:
      1. Name (text)
      2. Email (email)
      3. Message (textarea)
   3. Validate email format and require all fields

## **Functionality Requirements (JavaScript)**

* DOM manipulation to:
  + Add, display, edit, and delete book entries
  + Handle form input and submission
  + Show/hide dropdowns or menus
* Use **event listeners** for clicks, form submissions, etc.
* Provide form validation and show messages/errors in the UI

## **Styling & Layout Requirements (CSS)**

### **Layout**

* **Sticky Navbar**: Remains at the top when scrolling
* **Footer**: Always at the bottom of the page
* **Floating Action Button (FAB)**:
  + Bottom-right corner of screen
  + On click, shows a dropdown menu
* **Dropdown Menus**:
  + Use position: absolute for positioning
  + Include at least two across the project

### **Responsiveness**

* Use **media queries** to make the site mobile-friendly

### **Form Styling**

* Style all input fields, labels, and buttons
* Make invalid fields visually stand out
* Add hover/focus styles for buttons and inputs

### **Visual Design**

* Choose and use a consistent **color palette**
* Use **CSS variables** for colors and font sizes
* Use readable fonts and consistent spacing
* Include hover states for buttons/links
* Use transitions (optional but encouraged)

### **Code Structure**

* Organize your CSS (group related styles) (optional)
* Use comments to separate sections (/\* Footer \*/, /\* Navbar \*/)

## **Bonus Challenges (Optional)**

* Create a light/dark theme toggle
* Add a basic search filter to the book list
* Add a success message when a form is submitted
* Use **localStorage** to persist the library data

## **Submission Checklist**

* All required pages are created
* Book list supports add/edit/delete
* Forms work and are validated
* DOM manipulation is used throughout
* CSS layout and styling requirements are met
* Data is saved using localStorage
* Code is well-structured and commented

## **Resources**

* [DOM Basics – MDN](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction)