**Library Management System**

**Index:**

1. Test Framework Overview

2. Test Execution Results

3. Requirements Gaps

4. Bugs

**1. Test Framework Overview**:

Please find the test framework project details:

**Project Folder Name**: LibraryManagementSystem

**ReadMe File**: Attached as part of the zip file for instructions

The Test framework is built to test Library Management system, which covers below scenarios like add, edit, delete and read books (CRUD operations):

1. Check only authorized user should be able to login to Application and make changes

2. User with invalid credentials should be displayed with appropriate error messages

3. User logins with blank details should be displayed with appropriate error messages

4. User Journey from login to add a book with valid data and check if the book is added to system by reading the book details from the grid

5. User Journey from login to add a book with invalid data and check if error messages are displayed appropriately (specific to each field like title, author etc)

6. User Journey from login to add a book with blank details and check if error messages are displayed appropriately

6. User Journey from login to edit a book with valid data and check if updates are reflected

7. User Journey from login to edit a book with invalid data and check for error messages

8. User Journey from login to delete a book and check the book name is not displayed post deletion

9. Check for appropriate Welcome message and Log Out in the Landing page (‘Book List’ page).

10. Check for redirection to Login Page when Log Out button is clicked

**Maven**:

Built the Test framework as Maven project which is used as a build automation tool to manage project dependencies like selenium, testNG, WebDriverManager can be provided using pom.xml by reducing manual configuration.

src/main/java 🡪 to store page objects and abstract classes (AddBookPage, EditBookPage.java)

src/test/java 🡪 to store test cases and base classes (LMSTest.java)

**TestNG**:

Used TestNG as testing framework for organizing, managing and executing automated web tests.

This enables to use features like annotations for test set up, data driven testing and parallel execution. Used TestNG assertions to check if the test result and either fail or pass the test based on the expected results matching with actual result.

**Global Properties File**:

Used properties file to store and drive information in key value pair format e.g. browser name.

**Page Object Design Pattern**:

Used page object design pattern where for each web page of application as separate class, storing all web elements locators and its action methods, with in that class which creates an Object repository to improve code reusability, maintainability and readability by separating the test logic from page specific details. Created different page classes like AddBookPage.java, BookList.java, EditBook.Java etc.

**Page Factory**:

Used Page Factory class which simplifies the process of creating Page Objects with Page Object Design Pattern, this automatically initializes the web elements using “@FindBy” annotation to locate elements on the web page.

**Abstract Class**:

Created an abstract class to store all reusable component which can be inherited and used in page object classes.

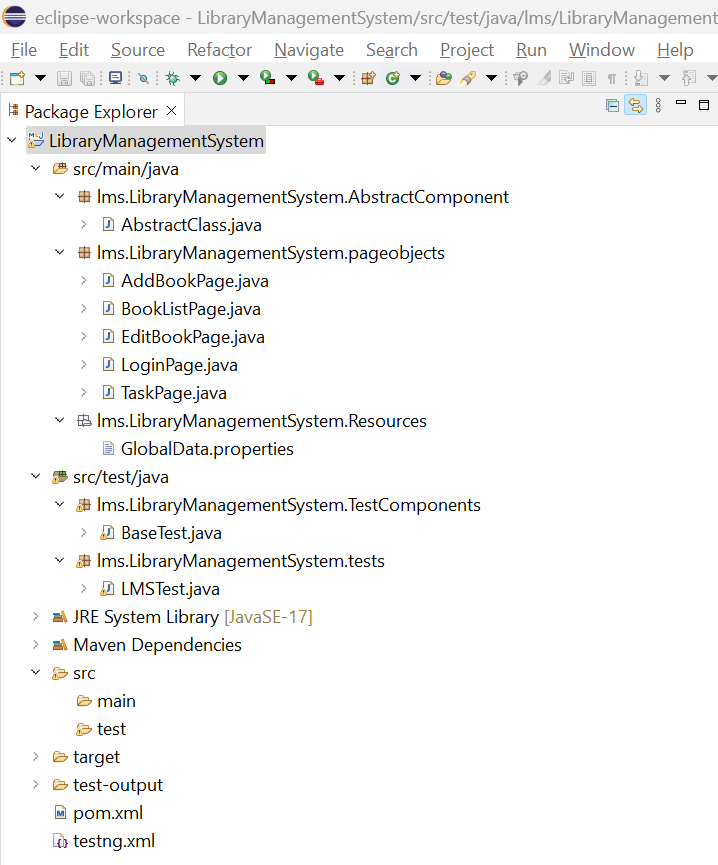
**Base Test Class**:

Created a Base Test Class to store all reusable methods which can be inherited and used in Test Class. Like teardown method to close or quit the browser.

**Tests**:

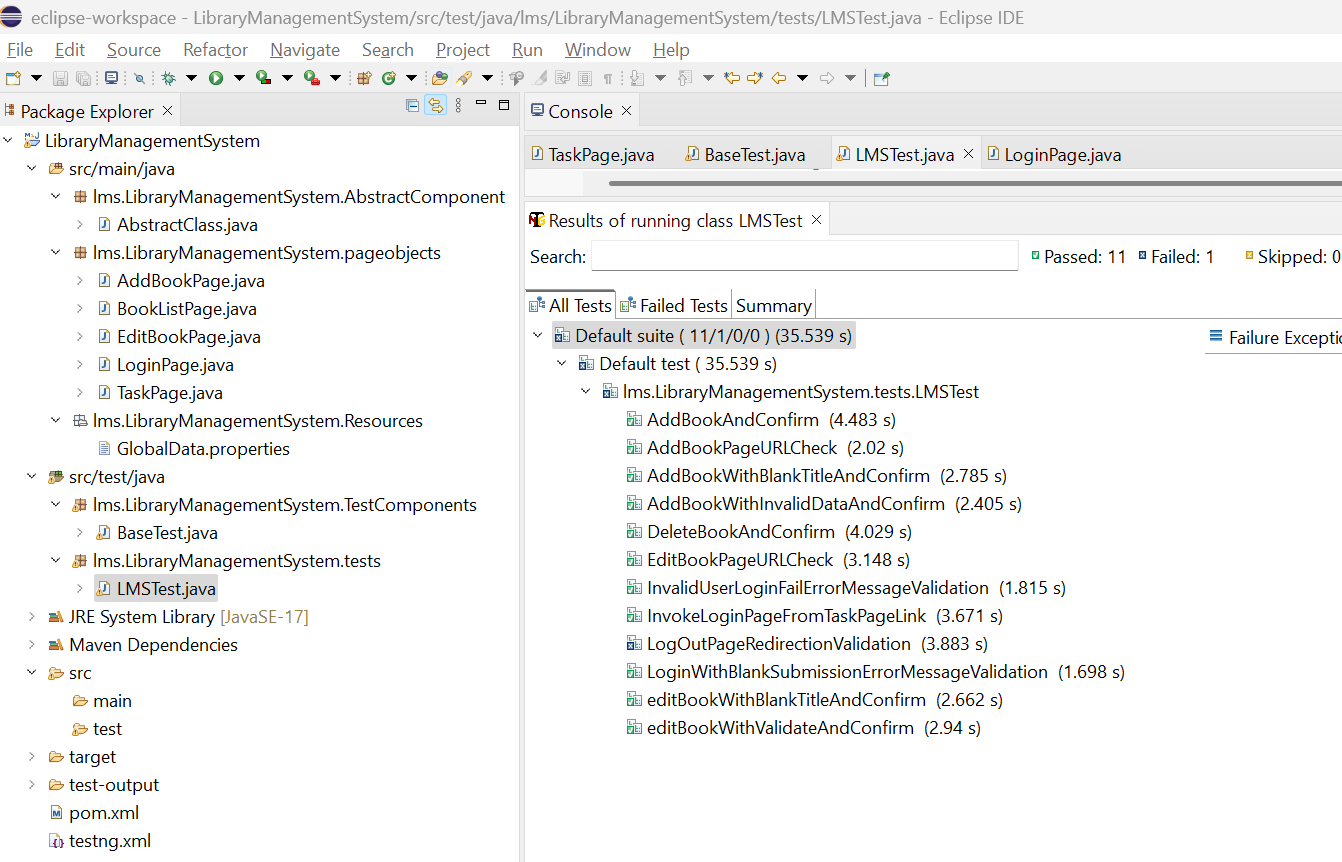
Written tests under src/test/test which inherits reusable methods from BaseTest.java class.

Project Folder Structure:

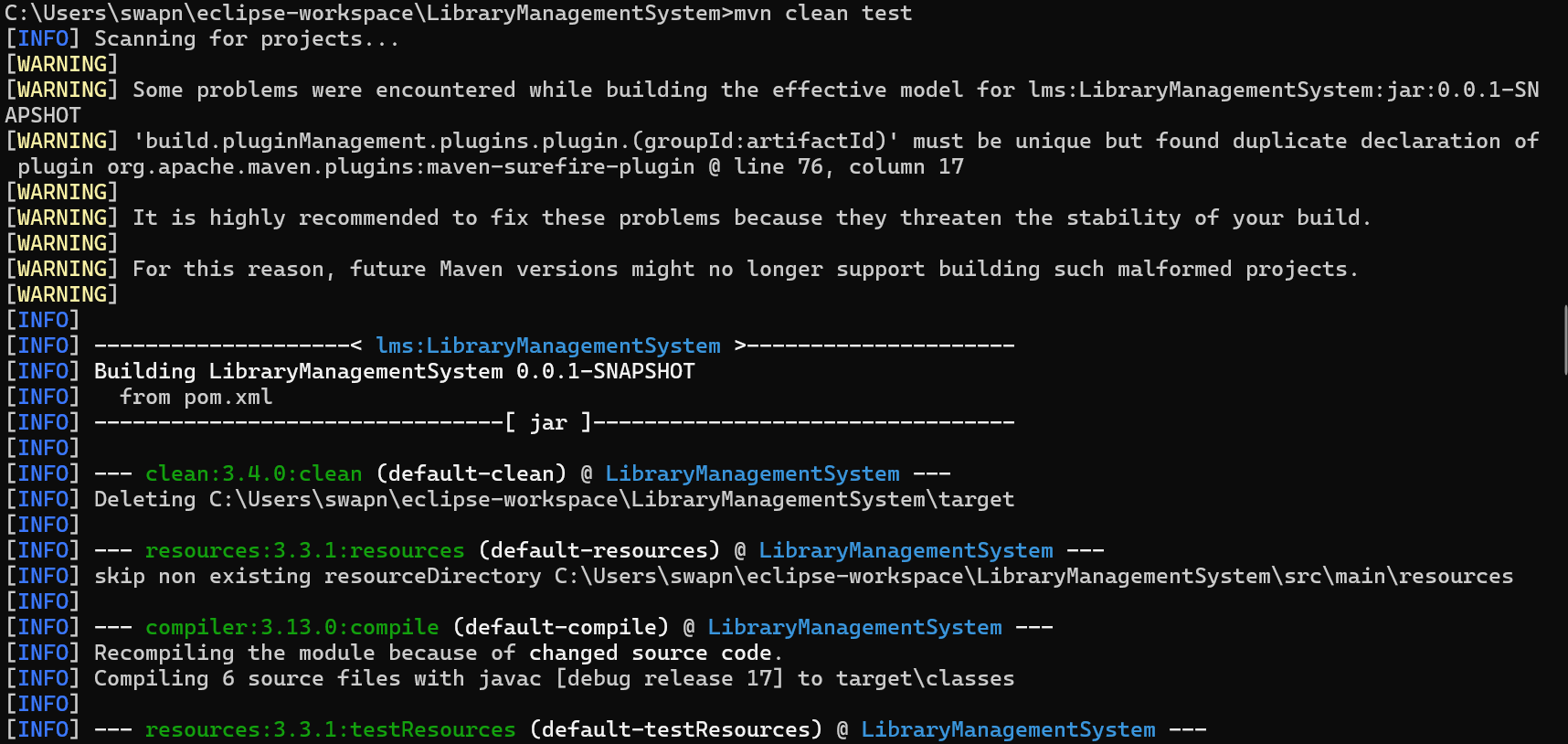
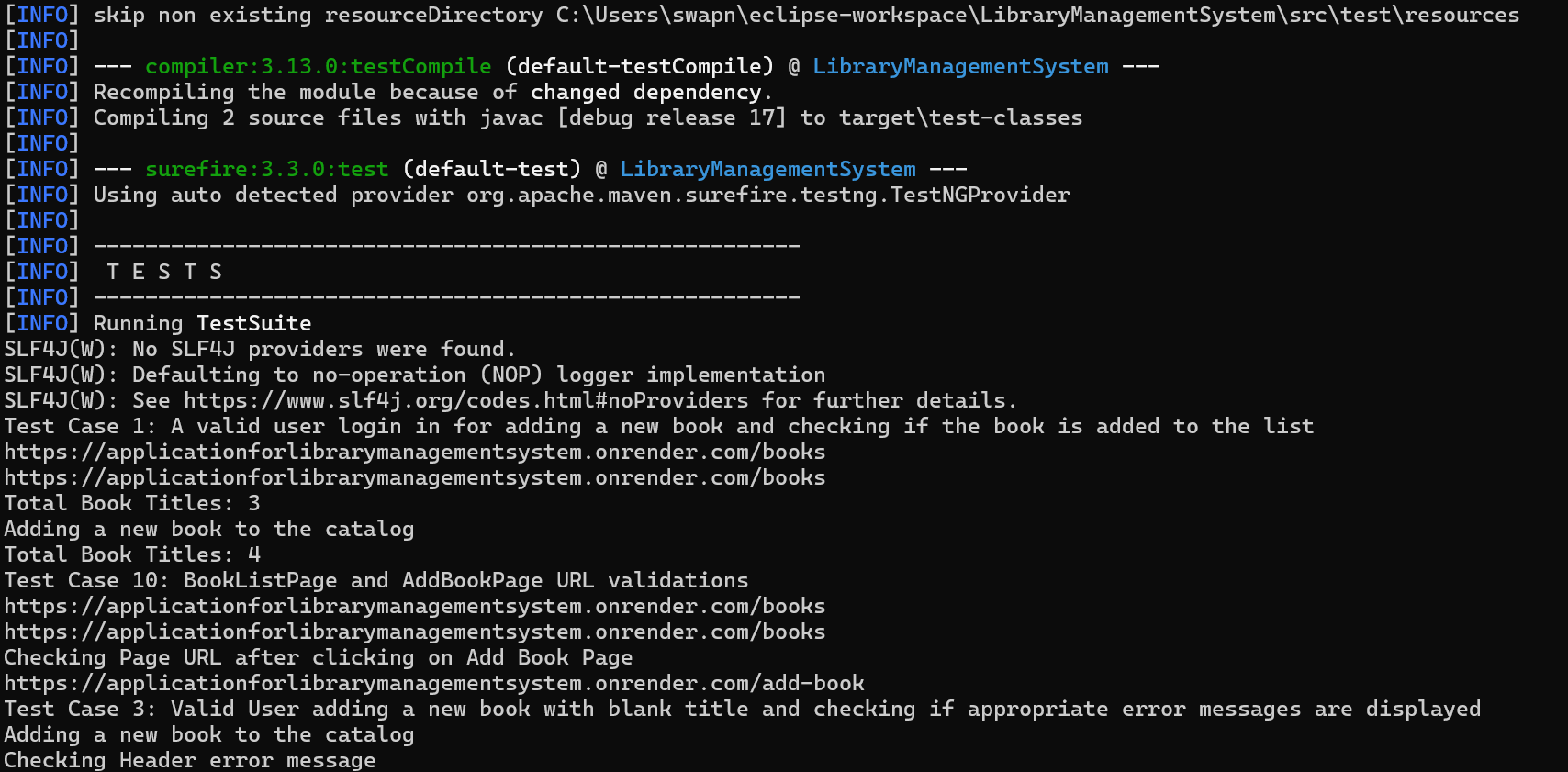
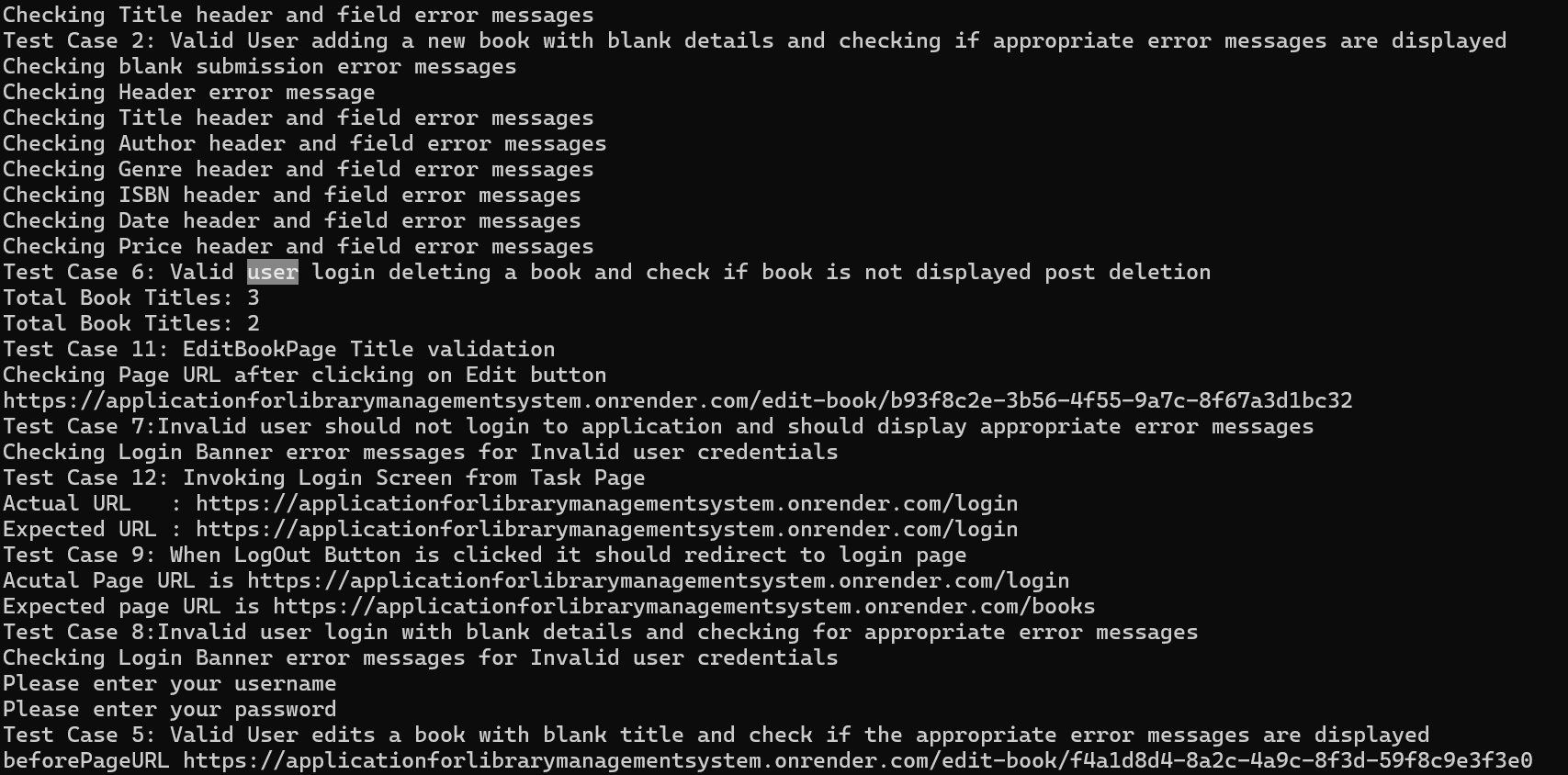
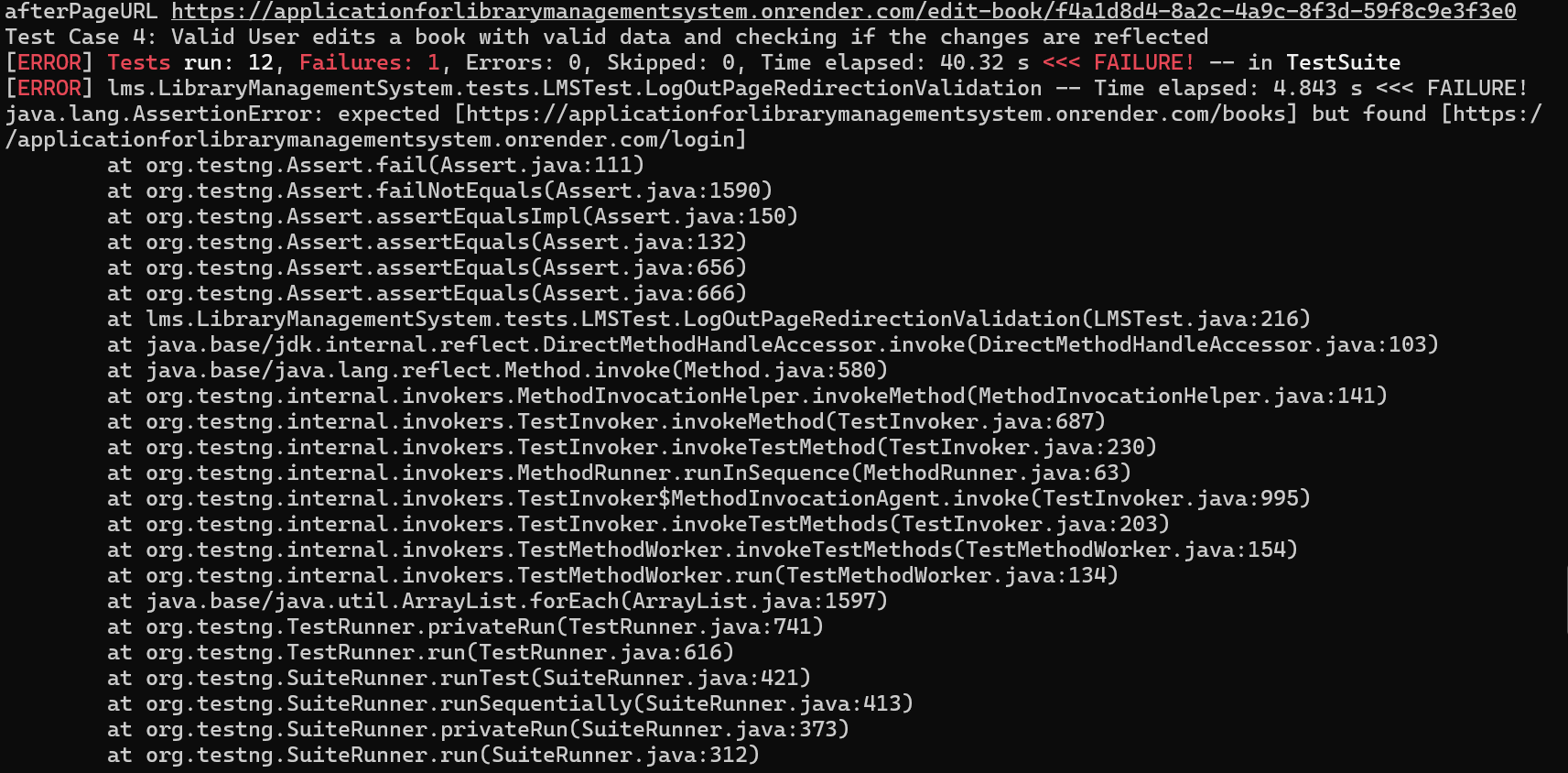
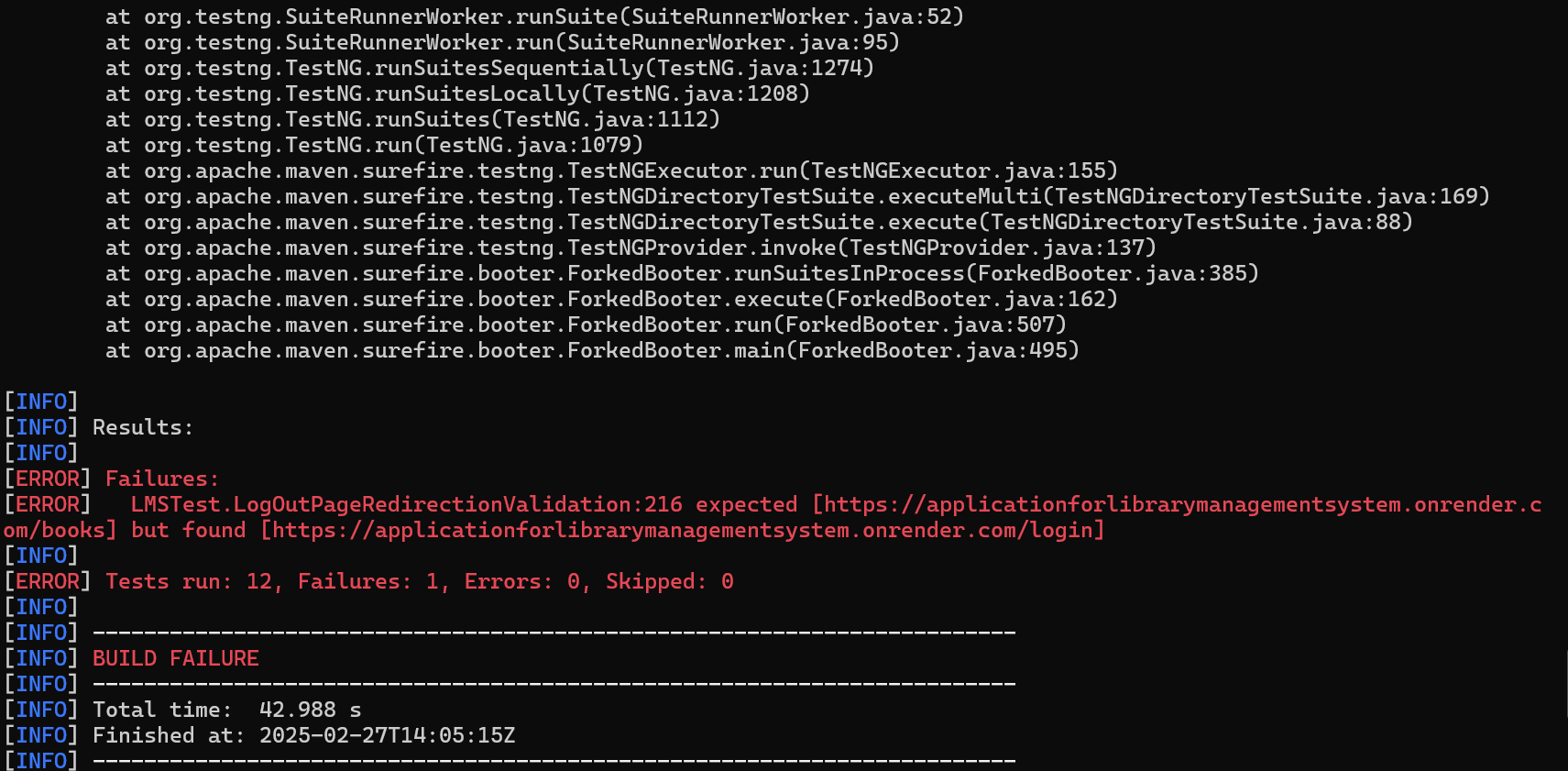


**2. Test Execution Results**:

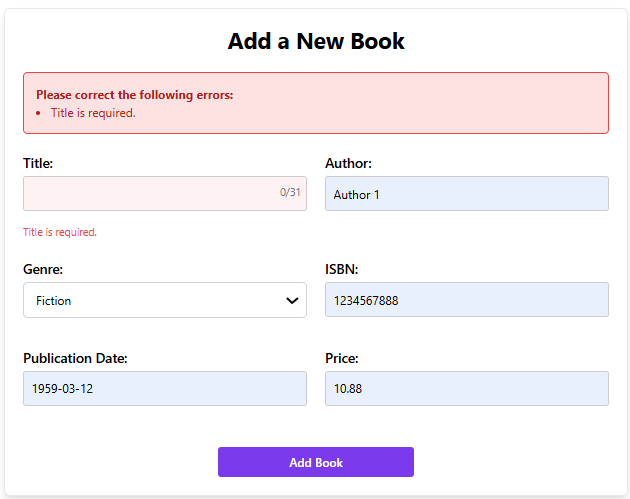
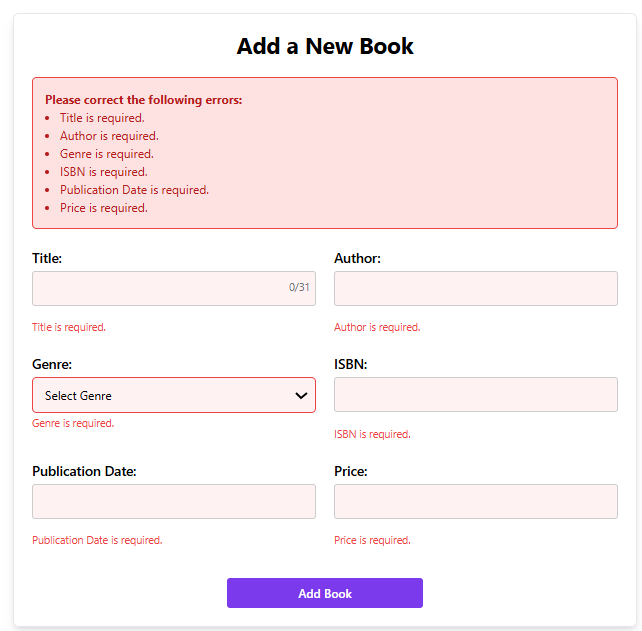
1 of 12 test cases is failing as the page redirection to Login Page is not happening

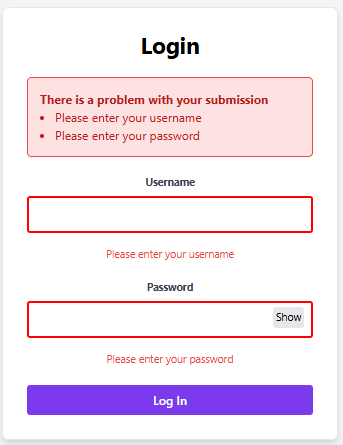
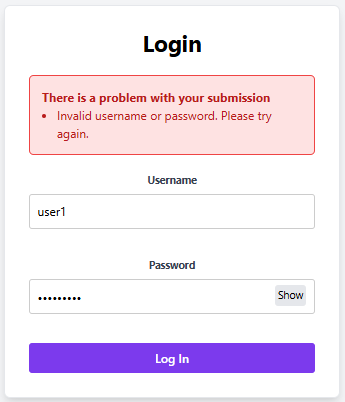
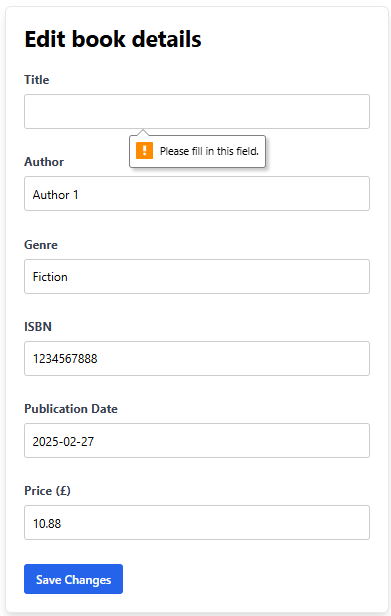


Execution through MVN:

Snippets of Error Validations Covered in the Automated Tests:





**3. Requirement Gaps**:

1. Search option should be provided to find a book when the book list grows

2. Previous and Next button are present and active even though there is only page to display. Previous button should be disabled/not be displayed when it is displaying first page. Next button should be either disabled/not be displayed when its displaying last page.

3. No message (“no books available to display”) is shown to the user when there are no books present in the system.

4. Book Genre is displayed as free text box in Edit operation; however, it is a dropdown for Add operation. Edit operation should also be let the user choose from dropdown for Genre selection.

5. Once ‘Add Book’ button is clicked on ‘Book List’ page, ‘Add a New Book’ page opens with details. However, there is no button option (either cancel or back) so that user can be taken back to ‘Add Book’ page.

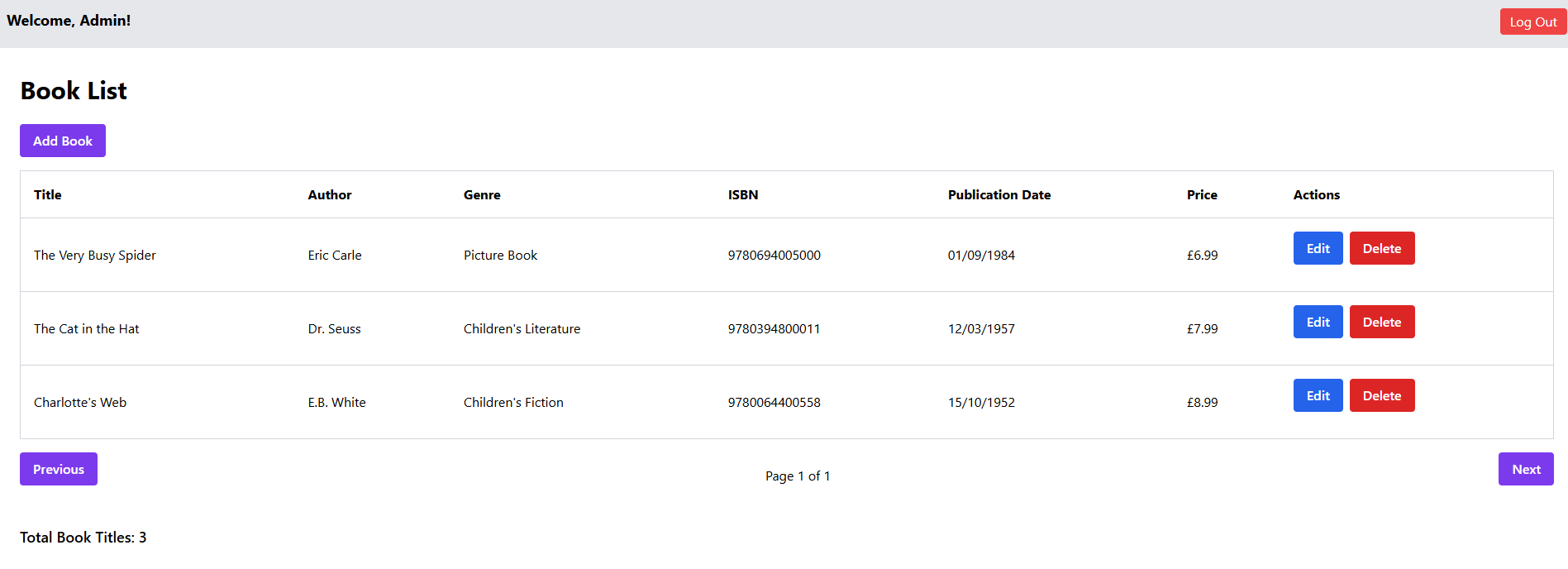
6. ‘Add a New Book’ page ‘Publication Date’ field is given as a free text which is allowing invalid data including text. A calendar date picker should be suggested which is be easy for selecting a date. Also, Date should be validated before submission and display appropriate error message.

7. Confirmation message should be displayed to the user, once Add, Edit, delete book operation is successfully completed

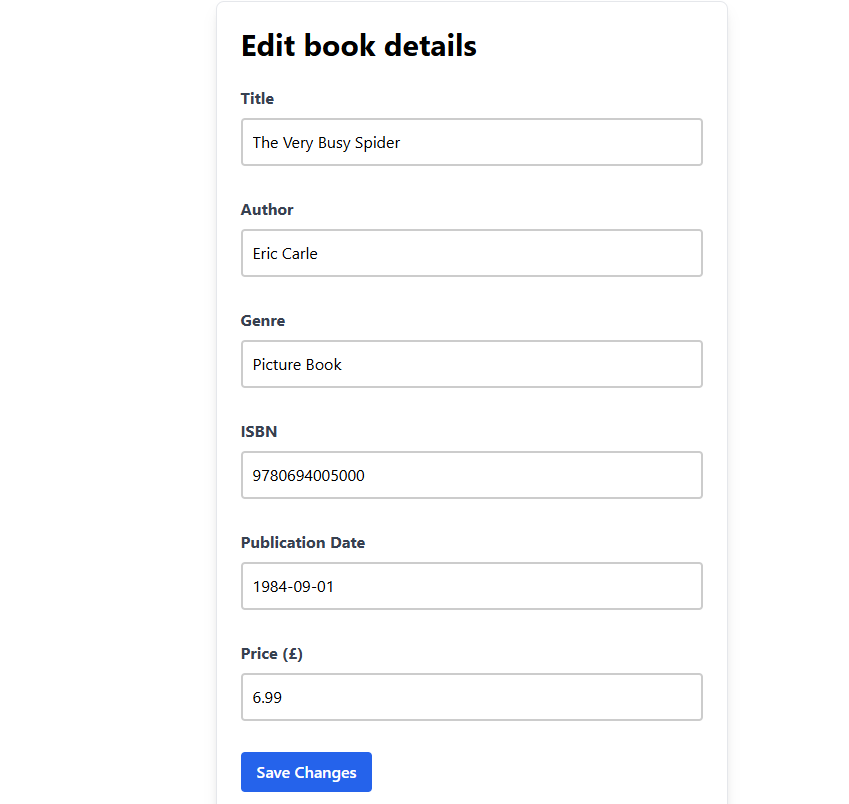
8. Application is allowing duplicates book details addition. This should be restricted.

Snippets for quick reference:

Below snippet is for requirement gaps 1,2,

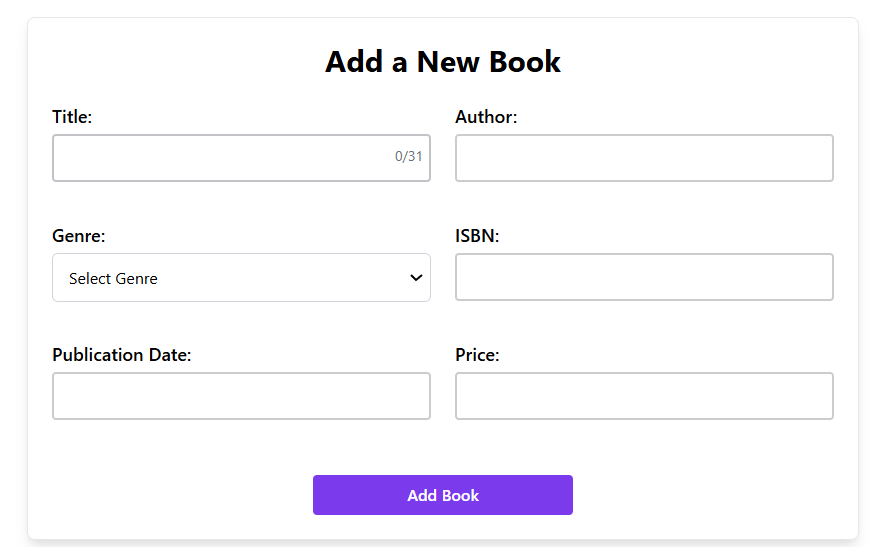


Below snippet is for requirement gaps 4 & 7 (Genre is free text instead of dropdown & no confirmation message)

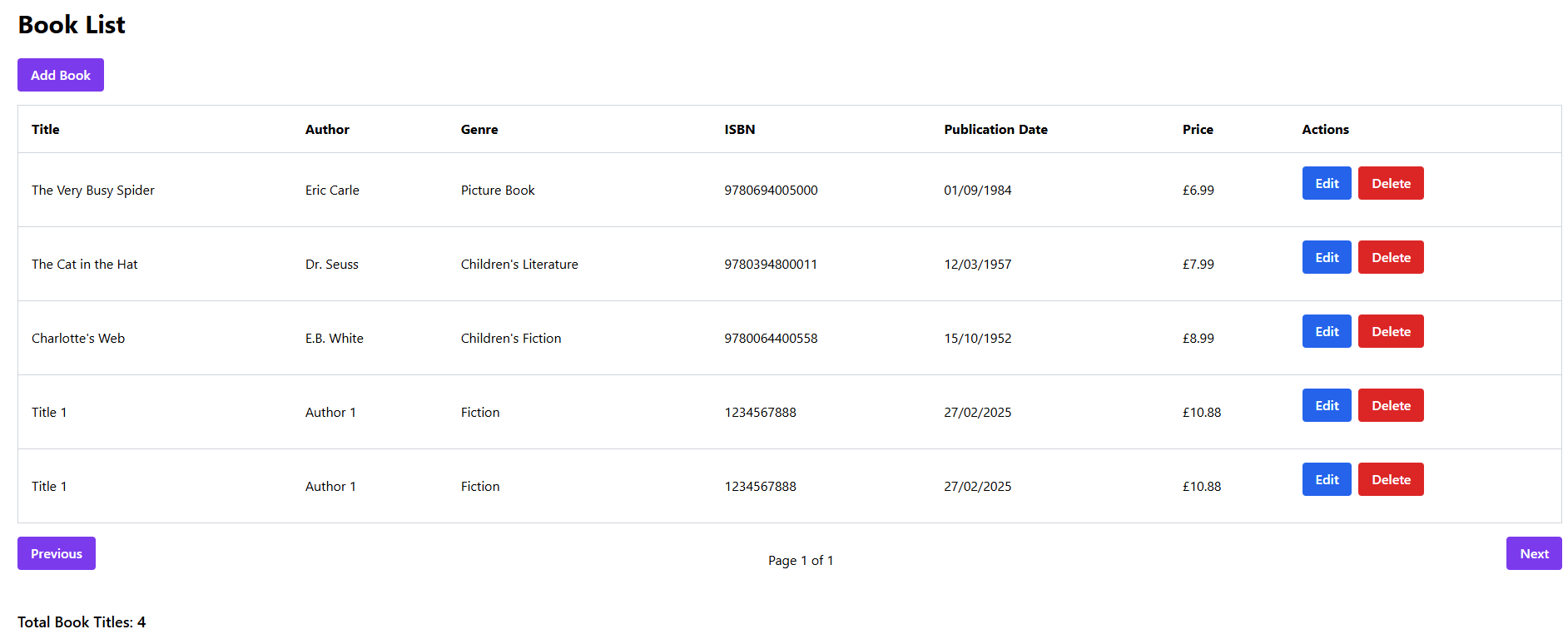


Below snippet is for requirement gaps 5 & 6 & 7

(No cancel button and use calendar date picker for Publication Date & no confirmation message)



Below snippet is for requirement gaps 8 – Allowing duplicates – but the total count is distinct



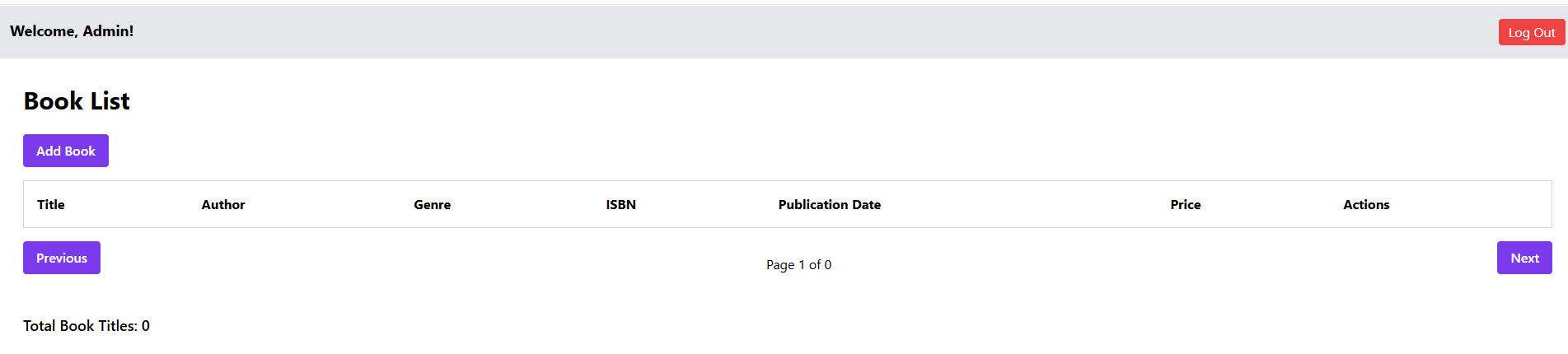
**4. Bugs**:

1. page 1of 0 where is there are no books to display (delete all books to check this scenario)

2. Add Book page is accepting invalid dates like text and displaying it as ‘Invalid Date’ in UI

3. Log Out action is not redirecting to Login Page and its still on Book List Page. 1 of the automated tests is failing as its application Issue (Test Case 9)

Snippet for requirement gap 3 & Bug 1 (no books for display message & page 1 of 0)



Name: Ravikumar Martha