Test Case ID	Test Scenario	Test Case Summary	Preconditions	Steps	Expected Result
TC_01	Verify User Registration	Verify User Registration form fields	The user registration form is implemented and accessible on the bookstore website.	1. Open Browser 2. Navigate to the user registration form	User should be able to navigate to the Registration page and should be able to click on text fields.
TC_02		Verify all text fields is displayed	The user registration form is implemented and accessible on the bookstore website.	1. Open Browser 2. Navigate to the user registration form	All text fields should be displayed such as Firstname, Lastname, Email, Password, Retype password, Phone no., Country etc
TC_03		Verify error messages in Registration page when user click on Submit button	The user registration form is implemented and accessible on the bookstore website.	1. Open Browser 2. Navigate to the user registration form 3. Enter invalid inputs or leave the mandatory field as blank	When user clicks on Submit button the error messages in red color should be displayed beside every mandatory field.
TC_04		Verify a password that does not meet minimum strength requirements (e.g., less than minimum length).	The user registration form is implemented and accessible on the bookstore website.	Fill invalid password	An error message should appear indicating the password strength requirements are not met.
TC_05		Verify by attempt to register with an email address that is already registered in the system.	The user registration form is implemented and accessible on the bookstore website.	Click on email field and try to enter an existing email.	An error message should appear indicating that the email address is already in use.
TC_06	Verify User Login	Test the valid credentials	The login form is implemented and accessible on the bookstore website.	1. Navigate to Login page 2. Enter the valid credentials 3. Click on login button	The user should be navigated to the Bookstore homepage.
TC_07		Test the invalid credentials	The login form is implemented and accessible on the bookstore website.	1. Navigate to Login page 2. Enter the invalid credentials like wrong password, or unregistered email 3. Click on login button	-The user should remain on the same login page. -Appropriate error messages should be displayed.
TC_08		Verify when user already Logged in and close the browser/tab, then reopen the application and navigate back to the site.	The login form is implemented and accessible on the bookstore website.	1. Navigate to Login page 2. Enter the valid credentials 3. Click on login button	User should remain logged in without having to re-enter credentials
TC_09		Verify by entering 3 invalid passwords	The login form is implemented and accessible on the bookstore website.	1. Navigate to Login page 2. Enter the invalid credentials like wrong password 3. Click on login button	The user account should be temporarily locked.
TC_10		Verify after logging in user leaves the browser/Page	The login form is implemented and accessible on the bookstore website.	1. Navigate to Login page 2. Enter the valid credentials 3. Click on login button	The user should be logged out automatically after 30 seconds if there is no activity by the user.
TC_11	Verify User Dashboard	Verify all relevant user information is displayed correctly on the dashboard. (e.g., name, email, profile picture and About bookstore).	User roles (e.g., admin, bookkeeper, regular user) should be defined and assigned correctly.	1. Navigate to Login page 2. Enter the valid credentials 3. Click on login button	User information should be accurately displayed as per the logged-in user's details on Homepage.
TC_12		Verify that recent activity or notifications are displayed in a timely and accurate manner.	User roles (e.g., admin, bookkeeper, regular user) should be defined and assigned correctly.	1. Navigate to Login page 2. Enter the valid credentials 3. Click on login button	Recent activities or notifications should be displayed chronologically and reflect the latest updates.

TC_13		Verify boundary conditions for data display (e.g., large number of notifications, lengthy text in messages).	User roles (e.g., admin, bookkeeper, regular user) should be defined and assigned correctly.	1. Navigate to Login page 2. Enter the valid credentials 3. Click on login button	The dashboard should handle large data without any issues.
TC_14		Verify navigation back to the dashboard after performing actions (e.g., editing profile, saving settings).	User roles (e.g., admin, bookkeeper, regular user) should be defined and assigned correctly.	1. Navigate to Login page 2. Enter the valid credentials 3. Click on login button 4. Edit required fields and navigate	User should be redirected back to the dashboard with the updated information displayed.
TC_15		Verify when user try to navigate unauthorized sections or perform actions without proper permissions (e.g., accessing admin-only features).	User roles (e.g., admin, bookkeeper, regular user) should be defined and assigned correctly.	1. Navigate to Login page 2. Enter the valid credentials 3. Click on login button 4. Click on unauthorized sections	Access should be denied, and an appropriate error message should be displayed indicating insufficient permissions.
TC_16	Verify Book Management API	Verify adding a book with valid inputs	User has appropriate permissions to add books. Use Postman to send requests	1. Send a POST request to add a book. 2. In body send title, author etc all mandatory fields	-HTTP status code 201 (Created) should be returned. -The response should contain the newly added book object with a unique identifier (e.g., book ID). -Verify that the book details match the input data sent.
TC_17		Verify adding a book with invalid inputs	User has appropriate permissions to add books. Use Postman to send requests	1. Send a POST request with missing mandatory fields (e.g., title or author) in body.	-The API should return a 400 (Bad Request) status code. -Verify that the response includes specific error messages indicating the missing fields.
TC_18		Verify retrieving book with valid inputs	User has appropriate permissions to add books. Use Postman to send requests	Send a GET request with valid id which is generating in response of POST request	-HTTP status code 200(OK) should be returned. -The response should contain the details of the book matching the provided ID. -Verify that the returned book details are correct and match the expected data.
TC_19		Verify retrieving a book with valid inputs using different user roles (e.g., Book admin, Bookkeeper, searcher).	User has appropriate permissions to add books. Use Postman to send requests	Send a GET request with valid id which is generating in response of POST request	-Users with appropriate permissions (e.g., admin, searcher) should be able to retrieve book details successfully. -Users without sufficient permissions should receive a 403 (Forbidden) status code or appropriate error message.
TC_20		Verify retrieving book with invalid Book ID inputs	User has appropriate permissions to add books. Use Postman to send requests	Send a GET request with invalid id.	-The API should return a 404 (Not Found) status code. -Verify that the response includes an error message indicating the book ID was not found.
TC_21		Verify Updating Book Details with valid inputs	User has appropriate permissions to add books. Use Postman to send requests	Send a PUT or PATCH request with valid id and updated book details in body.	-HTTP status code 200 (OK) or 204 (No Content) should be returned upon successful update. -Verify that the updated book details match the input data sent. -Retrieve the updated book details to confirm changes.

TC_22		Verify Updating Book Details with invalid inputs	User has appropriate permissions to add books. Use Postman to send requests	Send a PUT or PATCH request with a non-existent book ID.	-The API should return a 404 (Not Found) status code. -Verify that the response includes an error message indicating the book ID was not found.
TC_23		Verify deleting a book with valid user.	User has appropriate permissions to add books. Use Postman to send requests	Send a DELETE request with valid user permission	-Users with appropriate permissions (e.g., Store admin) should be able to delete books successfully. -HTTP status code 204 (No Content) should be displayed
TC_24		Verify deleting a book with valid inputs.	User has appropriate permissions to add books. Use Postman to send requests	Send a DELETE request.	-HTTP status code 204 (No Content) should be returned upon successful deletion. -Verify that the book is removed from the database. -Attempt to retrieve the deleted book should result in a 404 (Not Found) status code.
TC_25		Verify deleting a book with invalid inputs.	User has appropriate permissions to add books. Use Postman to send requests	Send a DELETE request with a non-existent book ID	-The API should return a 404 (Not Found) status code. -Verify that the response includes an error message indicating the book ID was not found.