

My Library (Test Plan)

Project Name: My Library

Date: 29.07.2025

Prepared by: Nethsilu Marasinghe

Content

1.0 INTRODUCTION	3
• OBJECTIVE	3
• PROJECT DESCRIPTION	3
• PROCESS TAILORING	3
• TECH STACK	3
2.0 ASSUMPTIONS/DEPENDENCIES	4
3.0 TEST REQUIREMENTS	4
4.0 TEST TOOLS	5
5.0 RESOURCE REQUIREMENTS	5
6.0 TEST SCHEDULE	5
7.0 RISKS/MITIGATION	6
8.0 METRICS	6

1.0 INTRODUCTION

1.1 OBJECTIVES

This Test Plan outlines the testing strategy for My Library, a personalized book management system designed to help users efficiently organize their personal book collections. The application enables users to seamlessly add, update, and delete book records, while maintaining an organized inventory of their literary assets. The system addresses common challenges faced by book enthusiasts, offering an intuitive, user-friendly interface to enhance cataloging and interaction with their collections.

The plan covers unit testing, integration testing, and system testing to ensure the system meets all functional and performance requirements.

1.2 PROJECT DESCRIPTION

My Library is an intuitive book management platform that:

- Allows users to effortlessly catalog and organize their personal book collections.
- Provides a clear overview of their entire library, including book and details.
- Offers a user-friendly interface for seamless book collection management.

1.3 TECH STACK

- Frontend: React.js + Tailwind CSS
- Backend: Node.js + Express.js
- Database: MongoDB Atlas

1.4 PROCESS TAILORING

- Testing Types: Functional, Error handling, API.
- Exclusions: GUI testing, API testing and compatibility testing beyond Windows OS.

2.0 ASSUMPTIONS / DEPENDENCIES

Assumptions:

- The developer will complete the code by 1 month.
- Users will enter data in the correct format.

Dependencies:

- Functional code delivery by development team.
- Availability of test environment (Windows OS).

3.0 TEST REQUIREMENTS

Functional Requirements to Test:

Book Collection Management

- Add, edit, and delete book entries with details (title, author, genre, publish year, etc.)

Library Tracking & Reporting

- Real-time display of total books, categorized by title, author and publish year.

4.0 TEST TOOLS

- Manual Testing (Other unit tests)
- Postman (API testing)
- OS & Browsers (Windows 10, Chrome, Firefox).
- Test Data (Sample data).

5.0 RESOURCE REQUIREMENTS

Role	Effort (Days)
Tester	1 Day
Development Support	1 Day

6.0 TEST SCHEDULE

Phase	Duration
Unit Testing	6 hours
Integration Testing	3 hours
System Testing	2 hours
API Testing	2 hours

7.0 RISK / MITIGATION

Risk	Mitigation
Performance > 1 minute	Optimize code or adjust requirements.

8.0 METRICS

- **Defects found per phase** (Unit/Integration/System).
- **Test coverage** (requirements traced to test cases).