

IT Project Management (IT3040)

Assessment on Tools

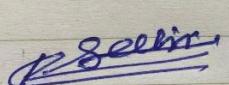
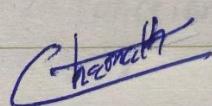
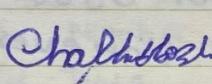
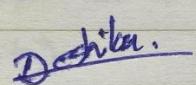
Group Number	Y3S1-WD-22	
Name	Registration No	Signature
Sachintha Nirmal S M	IT22234834	
Arthanayaka C T H	IT22251800	
Wijerathna C D W	IT22123268	
Vidurangi H I D G	IT22143990	

Table of Contents

Contribution to the project	1
Azure Board.....	2
GitHub.....	11
SonarQube.....	17
Selenium	19

Contribution to the project

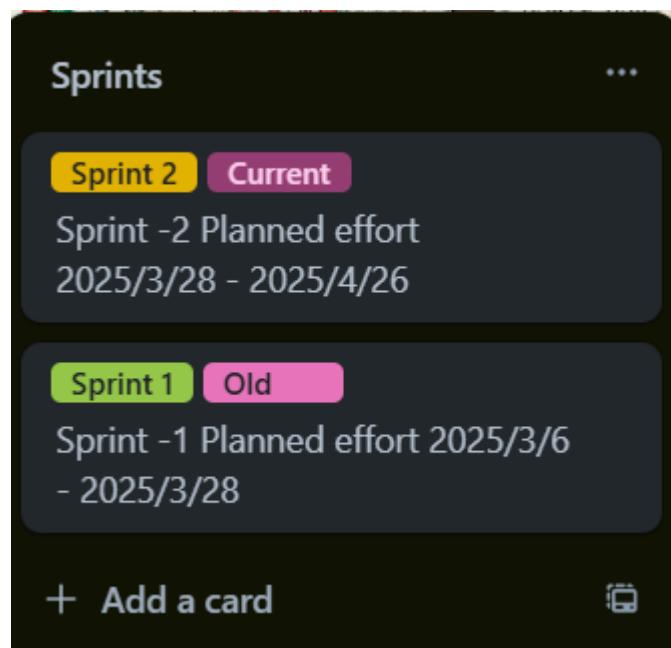
Student Reg. No	Responsible Tasks
IT22234834	I need to implement the user-end functionalities of the library and recycling management system like secure sign-up and login, easy membership renewal along with reminders, and accurate calculation of membership fees. I will also add reminders and a feedback system for enhancing user activity. I will implement a quiz feature that can be accessed only after login, to help recommend suitable books to members, thereby enhancing the interactivity and personalization of the system.
IT22251800	The admin is responsible for maintaining and organizing the E-Library collection by uploading, categorizing, updating, or removing digital books. They manage user accounts, oversee book ratings, and monitor member activity through detailed reports to ensure smooth operation and content relevance. Library members are responsible for engaging with the E-Library by browsing, reading, and rating books. They can manage their personal reading experience by using features like "Watch Later," search, reading progress tracking, and the "My Library" section to monitor and continue their reading journey efficiently.
IT22123268	The librarian manages book borrowing and returning through a dashboard, handling transactions, QR-based returns, record updates, fines, and data filtering, while tracking key stats like total and borrowed books. The admin oversees the full process by generating reports, analyzing trends, identifying overdues, and downloading analytics for decision-making
IT22143990	The librarian is responsible for maintaining an accurate and categorized book inventory by adding, updating, or removing books, and monitoring stock levels of available and borrowed items. They also manage fines by identifying overdue returns, calculating and applying charges, updating records, and adjusting fines when needed to ensure fair and transparent member accountability.

Azure Board

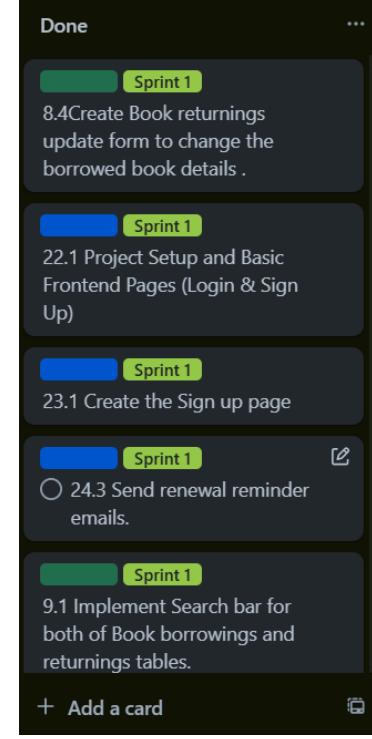
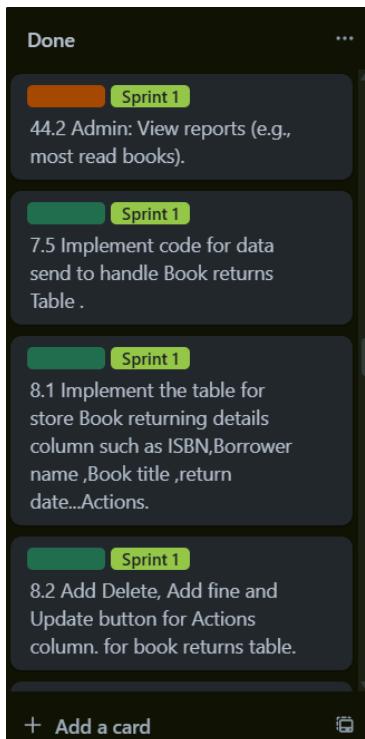
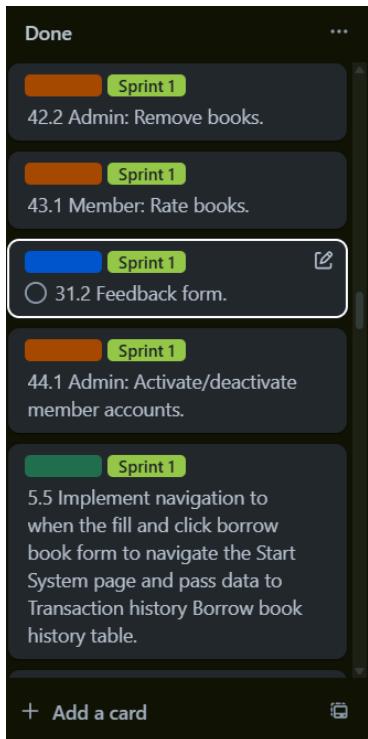
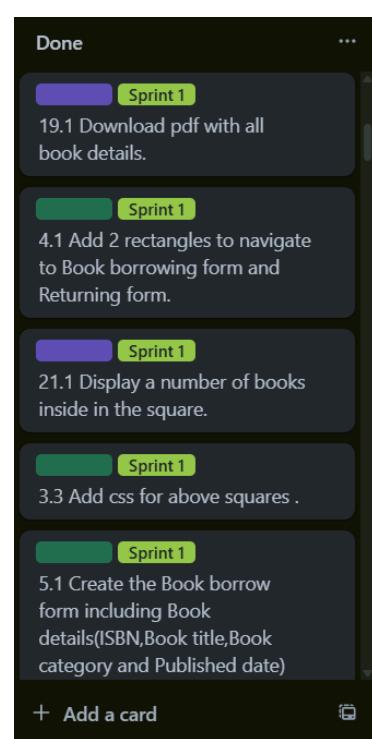
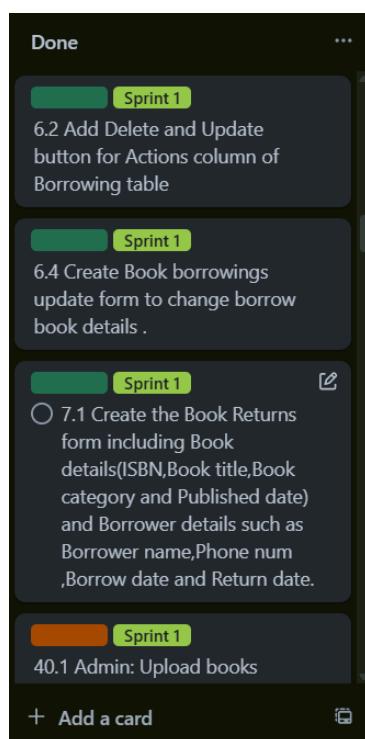
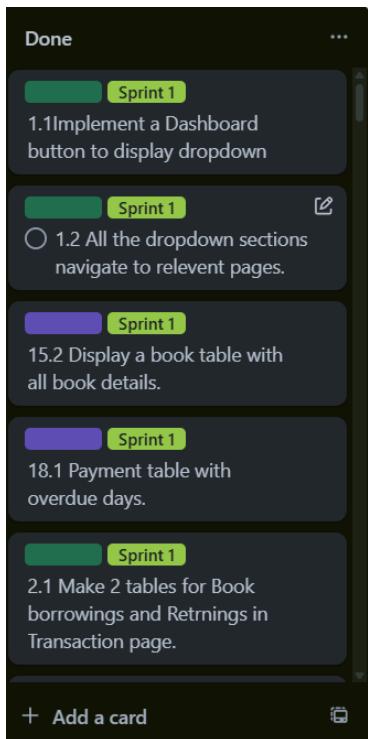
1. Work Items

Name in Azure Board	Student Registration No
Sachintha Nirmal	IT22234834
Chamath Arthanayaka	IT22251800
Chathubashini Wijerathna	IT22123268
Deshika Vidurangi	IT22143990

2. Sprints



3. Backlog items related to sprint 1



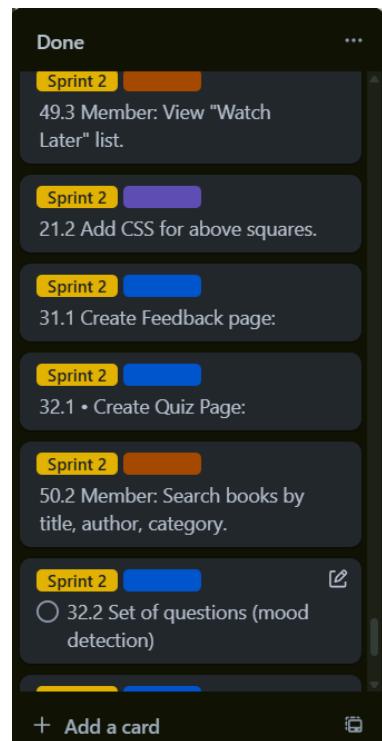
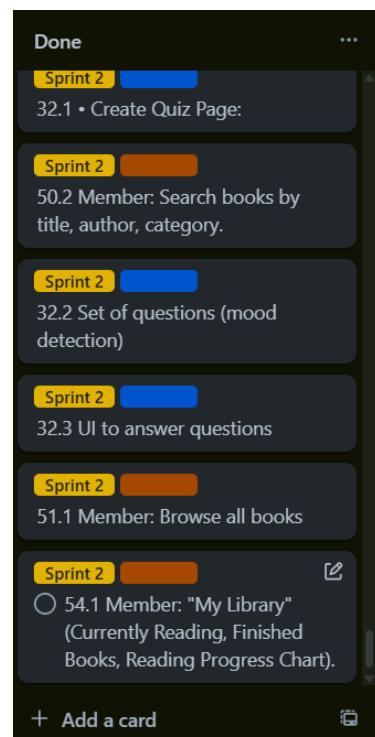
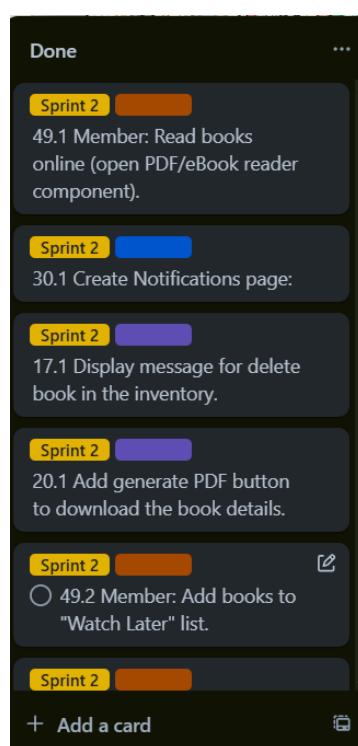
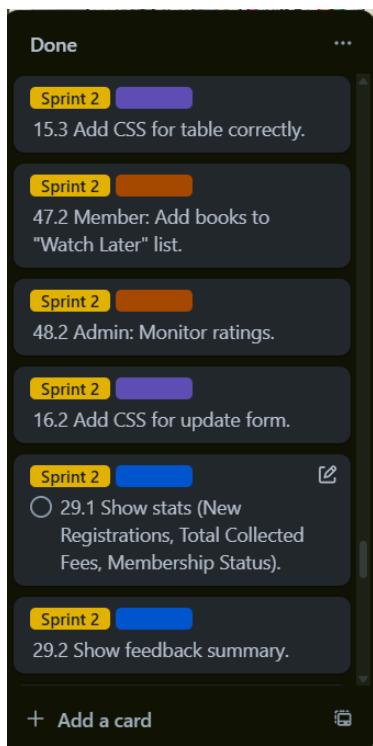
Sprint 1 Backlog Items:

- 41.1 Admin: Organize books into categories.
- 42. 1 Update book details.
- 7.2 Add Delete and Update button to Return form
- 7.3 Add validation for ISBN, Book name, Published date as Book details and Borrower name, phone number and Borrow date.
- 7.4 Implement navigation to when the fill and click return
- 24.3 Send renewal reminder emails.
- 9.1 Implement Search bar for both of Book borrowings and returnings tables.
- 9.2 Add filters to search book such as title, borrower name, category and date
- 11.1 Implement the Analyze button for Analyze borrow books table.
- 25.1 Create Membership Renewal page:

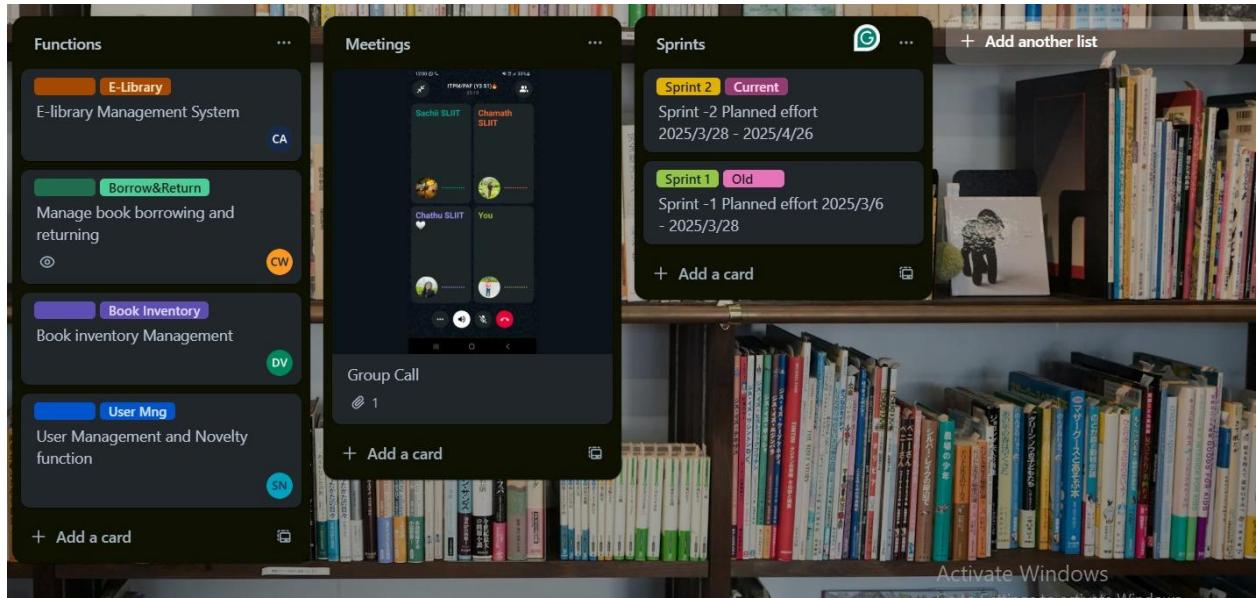
4. Backlog items related to sprint 2

Sprint 2 Backlog Items:

- 45.1 Admin: Activate/deactivate member accounts.
- 46.1 Member: Browse all books.
- 46.2 Member: Search books by title, author, category.
- 26.1 Membership Fee Calculation Frontend page create
- 11.2 Implement the Analyze page to visualize chart of the Book borrowings.
- 12.1 Implement tables for Book borrowing and returning transactions correctly display.
- 14.1 Implement code for download book borrowing details as a report
- 46.3 Member: View book details (title, author, description, pages, category).
- 47.1 Member: Read books online (open PDF/eBook reader component).
- 7.8 Add Automatically fine generate to Book returning handle Table for "Add fine" btn
- 15.3 Add CSS for table correctly.



5. Boards



trello.com/b/EbfxtBqY/library-management-system

The screenshot shows a Trello board titled "Library Management System". The board is organized into six columns: Product Backlog, Sprint Backlog, To do, In progress, Done, and Functions.

- Product Backlog:**
 - Borrow&Return [High]: 4. As a librarian, I want to start the system for book borrowing and returning, so that users can officially borrow and return books.
 - Borrow&Return [High]: 5. As a librarian, I want to handle borrowing books on behalf of users, so that I can assist users who are unable to complete the process themselves.
 - Borrow&Return [High]: 6. As a librarian, I want to manage borrowing details (delete, update), so that I can correct or remove incorrect records.
- Sprint Backlog:**
 - 3.1 Add three squares for visualize Total book borrowed books and available books in transaction page.
 - 3.2 Add Total books number correctly according to the Book inventory
 - 3.3 Add css for above squares .
 - 4.1 Add 2 rectangles to navigate to Book borrowing form and Returning form.
 - 4.2 Add css for above rectangles
- To do:**
 - 3.2 Add Total books number correctly according to the Book inventory
 - 17.2 Add good CSS for delete message.
 - 45.2 Admin: View reports (e.g., most read books).
 - 18.2 Create a payment table correctly.
 - 6.5 Implement pop up messages for update and delete functions for borrowing table
- In progress:**
 - 24.2 Create membership model (expiry date field).
 - 5.7 Implement backend codes for data send to handle Book borrows Table .
 - 6.3 Implement css code correctly for borrowings table
 - 48.3 Member: View 3 most recently added or updated books on homepage.
 - 7.6 Implement Backend code return book details update and
- Done:**
 - Sprint 1: 19.1 Download pdf with all book details.
 - Sprint 1: 4.1 Add 2 rectangles to navigate to Book borrowing form and Returning form.
 - Sprint 1: 21.1 Display a number of books inside the square.
 - Sprint 1: 3.3 Add css for above squares .
 - Sprint 1: 5.1 Create the Book borrow form including Book details(ISBN,Book title,Book category and Published date)
- Functions:**
 - E-library Management System
 - Manage book borrowing and returning
 - Book inventory Management
 - User Management and Novelty function

trellio.com/b/EbfxtbqY/library-management-system

Trello Workspaces Recent Starred Templates Create Beta Search Share ...

Library Management System

Product Backlog

- Borrow&Return High 1.As a librarian, I want a dashboard with a dropdown, so that I can easily navigate to different sections.
- Borrow&Return High 2. As a librarian, I want to view the transaction history of borrowed and returned books, so that I can keep track of all activities.
- Borrow&Return High 3. As a librarian, I want a visual overview showing total books, borrowed books, and available books in boxes, so that I can quickly monitor the library status.

Sprint Backlog

- 7.3 Add validation for ISBN, Book name, Published date as Book details and Borrower name, phone number and Borrow date.
- 7.4 Implement navigation to when the fill and click return book form to navigate the Start System page and pass data to Transaction history Return book history table.
- 7.5 Implement code for data send to handle Book returns Table.
- 7.6 Implement Backend code

To do

- 1.3 Make dropdown dashboard with good CSS
- 15.1 Make a add book form with good CSS.
- 41.4 Backend APIs to upload files, save book metadata, and manage categories.
- 26.5 Return correct fee via API backend
- 16.1 Create a display update message.

In progress

- 18.3 Make a card payment and cash payment.
- 2.2 Add CSS for both tables correctly.
- 53.1 Member: Track reading progress (percentage completed while reading).
- 19.2 Add CSS for the search bar.
- 5.3 Add dropdown list for Book category field to select relevant Book category.

Done

- Sprint 1 1.1 Implement a Dashboard button to display dropdown
- 1.2 All the dropdown sections navigate to relevant pages.
- 15.2 Display a book table with all book details.
- 18.1 Payment table with overdue days.
- 2.1 Make 2 tables for Book borrowings and Returns in Transaction page.

Functions

- E-library M...
- Manage b...
- Book inve...
- User Mana...

Activate Windows + Add a card Go to Settings to activate Windows.

27°C Mostly cloudy 11:35 PM 4/29/2025

trellio.com/b/EbfxtbqY/library-management-system

Trello Workspaces Recent Starred Templates Create Beta Search Share ...

Library Management System

Product Backlog

- High User Mng 22. As a user manager, I want to validate users' login credentials so that only authorized users can access the system.
- High User Mng 23. As a user manager, I want to allow new users to register (sign up) by providing required information so that they can create an account.
- High User Mng 24. As a user manager, I want to send membership renewal reminders to users before their membership expires so that they can renew on time.

Sprint Backlog

- 33.1 Create the login page and add CSS for the form.
- 33.2 Set up backend project, Create API for: • User Login (POST) • Connect frontend forms to backend APIs
- 34.1 Create the Sign up page
- 34.2 Create User Registration (POST) Backend
- 24.1 Provide "Renew" button if near expiration.
- 26.4 Develop fee calculation logic backend
- 29.3 Aggregate data: total members, fees, feedback types.
- 8.3 Implement pop up message to display when the click delete, update and Add fine button
- 30.4 Create notification model (message, userId, read/unread).

To do

In progress

- 48.1 Member: Rate books.
- 53.2 Member: View reading progress with a progress bar.
- 31.3 Submit feedback.
- 32.4 Show recommended book after quiz.
- 45.4 Frontend: Admin dashboard pages for member management and reports.

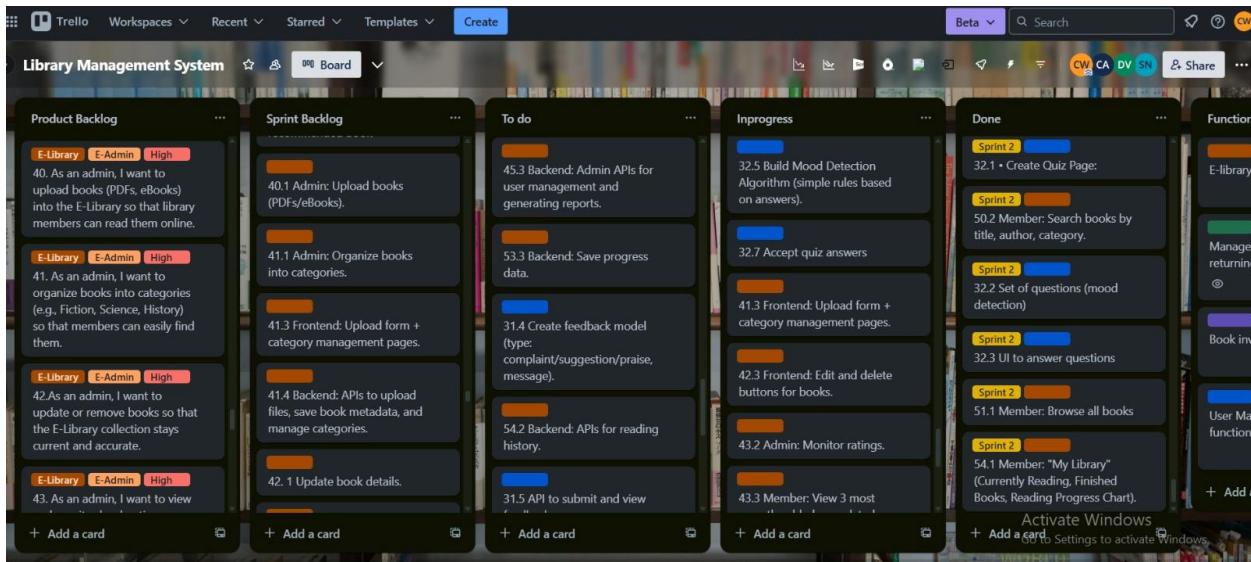
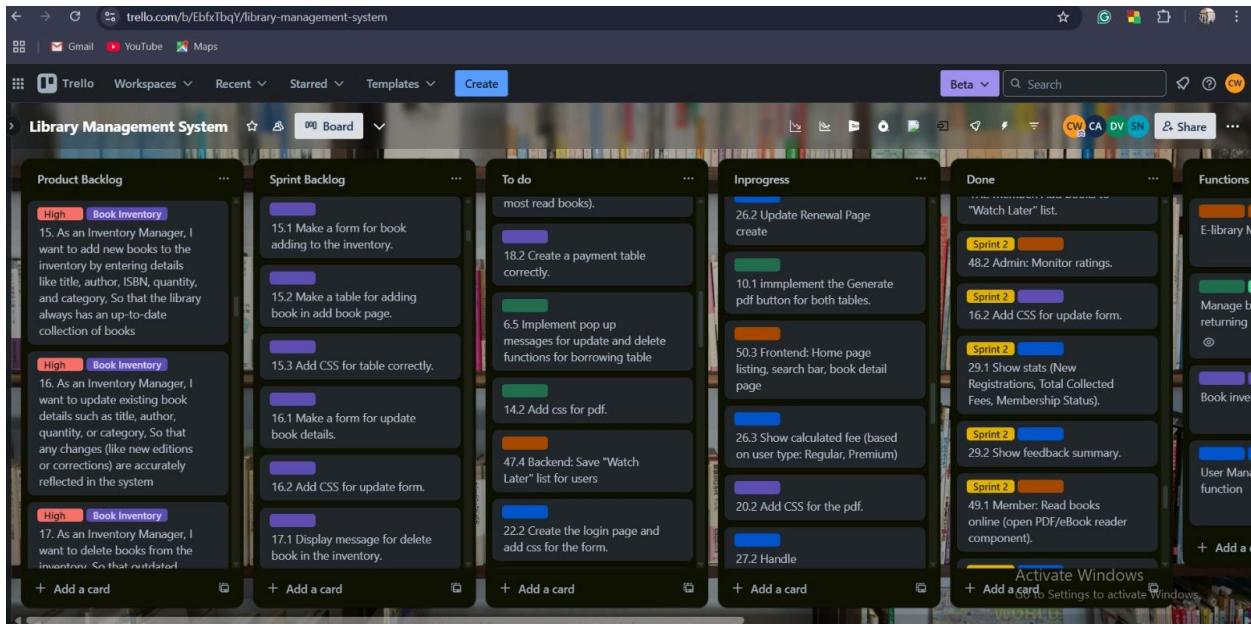
Done

- Later! list.
- Sprint 2 21.2 Add CSS for above squares.
- Sprint 2 31.1 Create Feedback page.
- Sprint 2 32.1 Create Quiz Page.
- Sprint 2 50.2 Member: Search books by title, author, category.
- Sprint 2 32.2 Set of questions (mood detection)
- Sprint 2 32.3 UI to answer questions

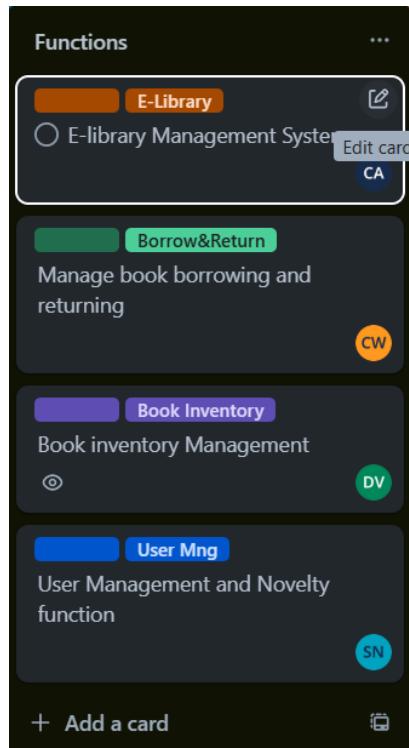
Functions

- E-library M...
- Manage b...
- Book inve...
- User Mana...

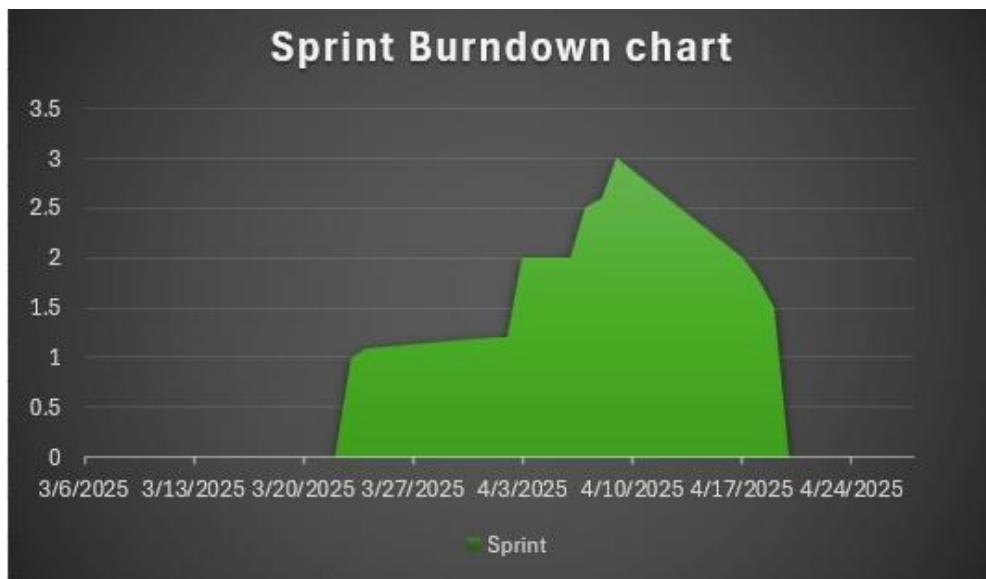
Activate Windows + Add a card Go to Settings to activate Windows.



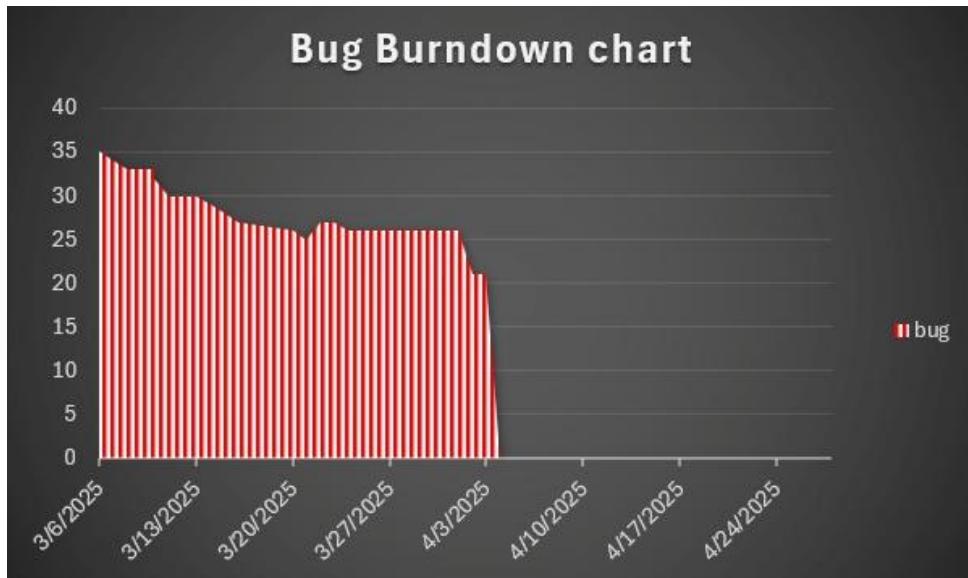
6. Feature



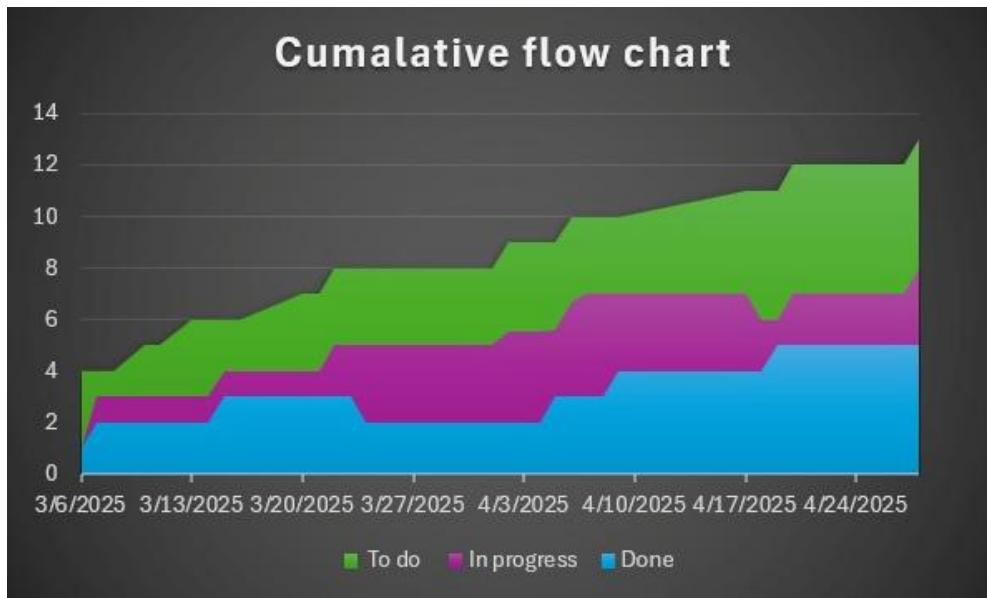
7. Sprint burndown chart



8. Bug burndown chart



9. Cumulative flow diagram



10. Trello Link - <https://trello.com/b/EbfxTbqsY/library-management-system>

GitHub

GitHub Link: <https://github.com/sachintha-nirmal-/Library-management-system.git>

1. Repository view

The screenshot shows the GitHub repository page for 'Library-management-system'. The repository is private and was created by 'sachintha-nirmal-sm'. It has 5 branches and 0 tags. The main branch was last updated on b4c206d · now. There are 33 commits. The repository contains files like 'backend', 'frontend', '.gitattributes', and '.gitignore'. A 'README' file is present, with a button to 'Add a README'. The 'About' section indicates no description, website, or topics provided. The 'Contributors' section lists four contributors: 'sachintha-nirmal-sm' (Sachintha Nirmal), 'Chamath-0027' (Chamath Tharinda), 'Deshika-Inventory-Management' (Deshika), and 'Shruthi-Inventory-Management' (Shruthi). The 'Releases' and 'Packages' sections are currently empty.

2. File setup

This screenshot is identical to the one above, showing the GitHub repository page for 'Library-management-system'. It highlights the 'Add a README' button, indicating the next step in the file setup process. The repository structure, commit history, and contributor information remain the same.

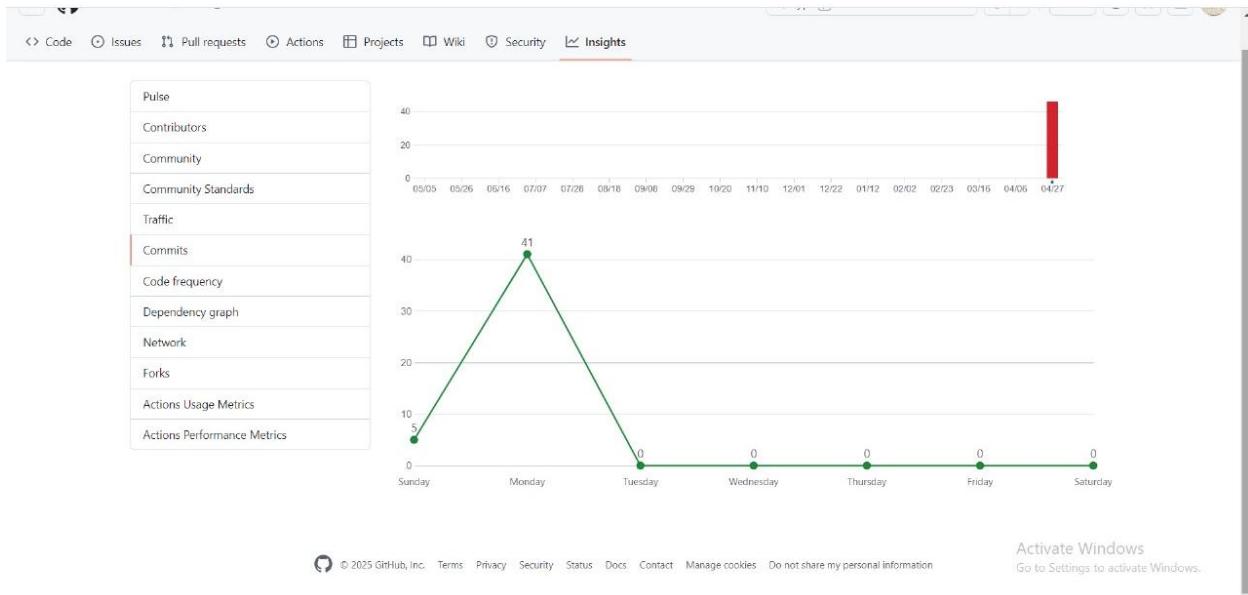
3. Branches

The screenshot shows a GitHub repository page for 'Library-management-system'. The repository is private. At the top, there are navigation links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. A search bar is at the top right. Below the header, there's a main content area with tabs for 'main' (selected), 'Branches' (which shows 5 branches), and 'Tags' (0 tags). A modal window titled 'Switch branches/tags' is open, showing a search bar and a list of branches: 'main' (checked), 'deshika_inventory-management', 'devmini_book-management', 'sachintha_user-management', and 'chamath_e-library'. There's also a link to 'View all branches'. To the right of the branches, there's a timeline of commits, an 'About' section with no description, and sections for Activity, Releases, Packages, and Contributors.

4. Contributors

GitHub Username	Student Registration No
Sachintha-nirmal-sm	IT22234834
Chamath-0027	IT22251800
Devmini26	IT22123268
deshikavdu	IT22143990

5. Commit graph

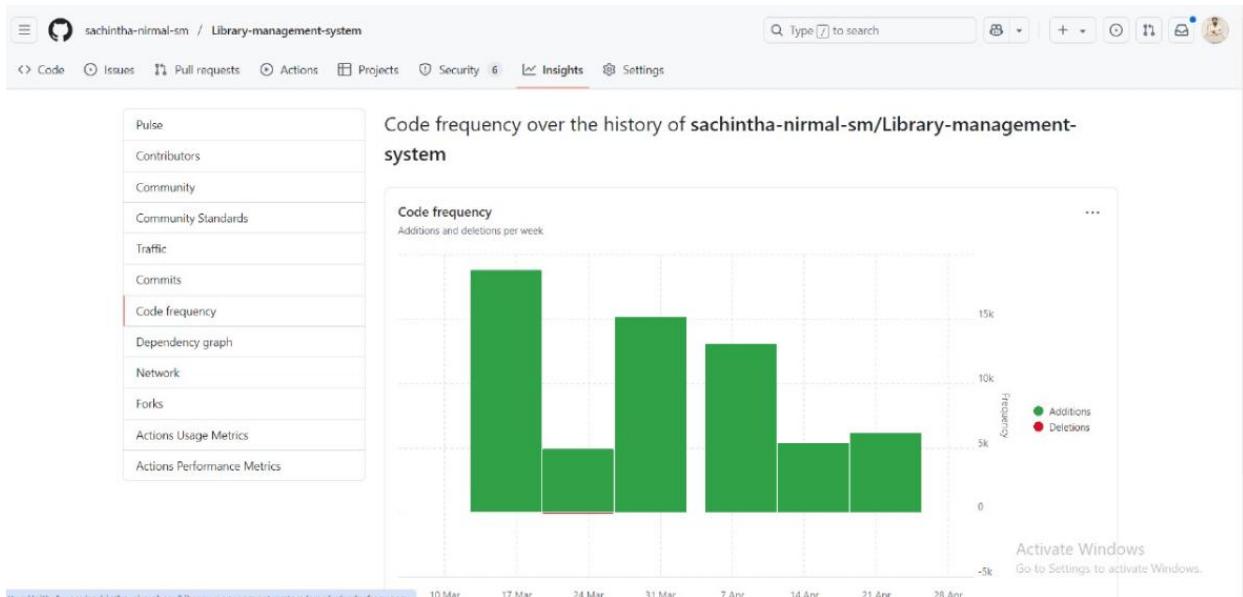


6. Commit History

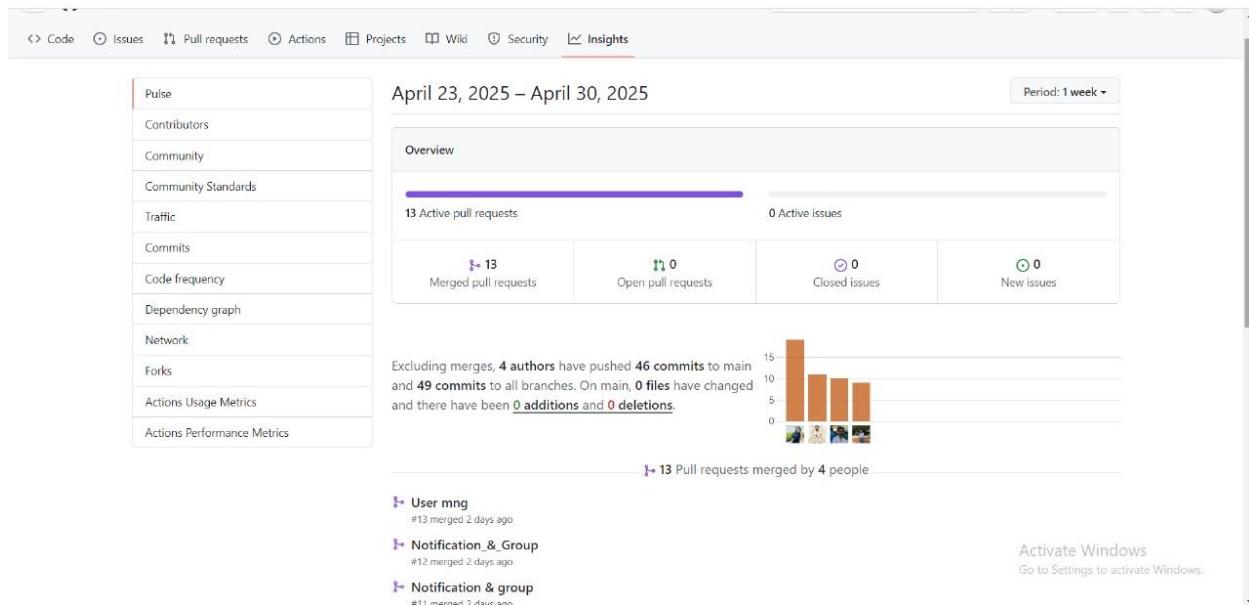
The figure shows a detailed commit history for the 'main' branch on March 25, 2025. The commits are listed in chronological order from top to bottom:

- Merge pull request #7 from sachintha-nirmal-sm/deshika_inventory-management (Verified) 46f6035
- Merge branch 'main' into deshika_inventory-management (Verified) 29b4c34
- i added the validations for bookform. (Verified) ab0f9be
- Merge pull request #6 from sachintha-nirmal-sm/chamath_e-library (Verified) 85dc5f9
- Merge branch 'main' into chamath_e-library (Verified) 1389078
- new book model , navigation ang png commit (Verified) 3883647
- footer commit (Verified) a26bd05
- app first commit (Verified) 0bd708d
- Merge pull request #5 from sachintha-nirmal-sm/sachintha_user-management (Verified) 9ac127b

7. Code Frequency



8. Pulse



9. ReadMe.md

The screenshot shows a GitHub repository page for a "Library Management System". The page includes a README file with a brief description of the system's features and a table of contents for its sections. On the right side, there are sections for releases, packages, contributors, languages, and suggested workflows.

README

Library Management System

A comprehensive MERN stack-based Library Management System that simplifies and streamlines library operations with modern digital solutions. This system offers robust features such as user management, inventory tracking, e-library access, and book lending/returning services. It also includes an innovative *Mood-Based Book Recommendation System* to enhance user experience.

Features

- User Management**
 - User registration, login, and profile management
 - Role-based access (e.g., admin, user)
- Inventory Management**
 - Add, update, delete books
 - Track stock levels and categories
- E-Library Management**
 - Access and read digital copies of books
 - Manage digital content and file uploads
- Book Borrowing & Returning**
 - Issue and return book records
 - Send due reminders and borrowing notifications

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 4

- sachintha-nirmal-sm Sachintha Nirmal
- Chamath-0027 Chamath Tharinda
- Devmini26 Devmini wijerathna
- deshikavida Deshika Vidurangi

Languages

- JavaScript 58.9%
- CSS 40.9%
- HTML 0.2%

Suggested workflows

Based on your tech stack

- Webpack Configure Build a Node.js project with npm and webpack
- Gulp Configure Build a Node.js project with npm and gulp.

The screenshot shows a GitHub repository page for a "Library Management System". The page includes a README file with instructions for setting up the project, a folder structure, and a contributing section. At the bottom, there is a license link.

README

```
cd backend
npm install
npm run dev
```

3. Frontend Setup:

```
cd frontend
npm install
npm start
```

4. Environment Variables: Create .env files in both frontend and backend folders with appropriate configuration (MongoDB URL, JWT secret, etc.)

Folder Structure

```
library-management-system/
├── backend/
│   ├── models/
│   ├── routes/
│   └── controllers/
└── frontend/
    ├── src/
    └── public/
```

Contributing Feel free to fork this repo and make contributions. Pull requests are welcome!

License MIT License

Mood-Based Book Recommendation

- Suggests books based on the user's inferred mood
- Enhances reading experience using emotion-based algorithms

Tech Stack

- Frontend: React.js, Tailwind CSS
- Backend: Node.js, Express.js
- Database: MongoDB with Mongoose
- Authentication: JWT
- Other: Cloudinary (for images/PDFs), RESTful APIs

Installation

- Clone the repository:
git clone https://github.com/sachintha-nirmal-sm/Library-management-system.git
- Backend Setup:
cd backend
npm install
npm run dev
- Frontend Setup:

Deno

Test your Deno project

Configure

More workflows

Dismiss suggestions

SonarQube

1. Overview status of analyzed projects

```
Administrator: Command Prompt
23:05:22.899 WARN * frontend/src/pages/WatchLater.css
23:05:22.902 WARN * frontend/src/pages/h-Learnmore.jsx
23:05:22.922 WARN * frontend/src/pages/Home1.jsx
23:05:22.929 WARN * frontend/src/pages/Home.css
23:05:22.929 WARN * backend/models/Book.js
23:05:22.930 WARN * frontend/src/pages/CategoryPage.jsx
23:05:22.932 WARN * frontend/src/pages/Home.jsx
23:05:22.933 WARN * frontend/src/pages/Login.css
23:05:22.934 WARN * frontend/src/pages/WatchLater.jsx
23:05:22.934 WARN * frontend/src/pages/h-Learnmore.css
23:05:22.937 WARN * frontend/src/pages/Home1.css
23:05:22.942 WARN This may lead to missing/broken features in SonarQube
23:05:22.943 INFO CPD Executor 3 files had no CPD blocks
23:05:22.944 INFO CPD Executor Calculating CPD for 8 files
23:05:22.954 INFO CPD Executor CPD calculation finished (done) | time=46ms
23:05:22.974 INFO SCM revision ID '38836e779ac03970e203866d28ce48186b7a9630'
23:05:24.847 INFO Analysis report generated in 395ms, dir size=465.3 kB
23:05:25.055 INFO Analysis report compressed in 183ms, zip size=133.2 kB
23:05:30.061 INFO Analysis report uploaded in 500ms
23:05:30.065 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=Library
23:05:30.072 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
23:05:30.078 INFO More about the report processing at http://localhost:9000/api/ce/task?id=b44597da-ed24-47db-a3f6-f44dc37eac6d
23:05:30.462 INFO Analysis total time: 43.806 s
23:05:30.464 INFO SonarScanner Engine completed successfully
23:05:31.012 INFO EXECUTION SUCCESS
23:05:31.015 INFO Total time: 49.372s

C:\library\library\Library-management-system>
```

The screenshot shows the SonarQube dashboard for the 'main' project. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and More. The main content area displays a summary card with the following information:

- 3.7k Lines of Code
- Version 1.0
- Last analysis 2 minutes ago
- Quality Gate: Passed
- Overall Code Status: Passed
- Open Issues: 56 (Security: 0, Reliability: 56, Maintainability: 119)
- Coverage: 0.0%
- Duplications: 1.6%
- Security Hotspots: 0

The Activity section below the summary card shows a message: "There isn't enough data to generate an activity graph." At the bottom of the page, a footer note states: "First analysis: 120 issues - 0.0% Coverage - 1.6% Duplications".

2. Project overview

The screenshot shows the SonarQube 'Projects' page. At the top, there are tabs for 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', 'Administration', and 'More'. A message at the top indicates that an embedded database is being used for evaluation purposes only. Below this, a note states that security, reliability, and maintainability counts and ratings have changed. The main area displays the 'LibraryManagementSystem' project, which is marked as 'PUBLIC' and has a 'Passed' status. It shows the last analysis was 5 minutes ago, involving 3.7k Lines of Code in CSS and JavaScript. The project has 0 security issues, 56 reliability issues, 119 maintainability issues, 0 hotspots reviewed, 0.0% coverage, and 1.6% duplications. On the left, there are filters for Quality (Passed, Failed), Security (severity levels: Info, Low, Medium, High, Blocker), Reliability (severity levels: Info, Low, Medium, High, Blocker), and Maintainability (severity levels: Info, Low, Medium, High, Blocker). A sidebar on the left lists 'My Favorites' and 'All' projects. At the bottom, there is footer information about SonarQube technology and a build ID.

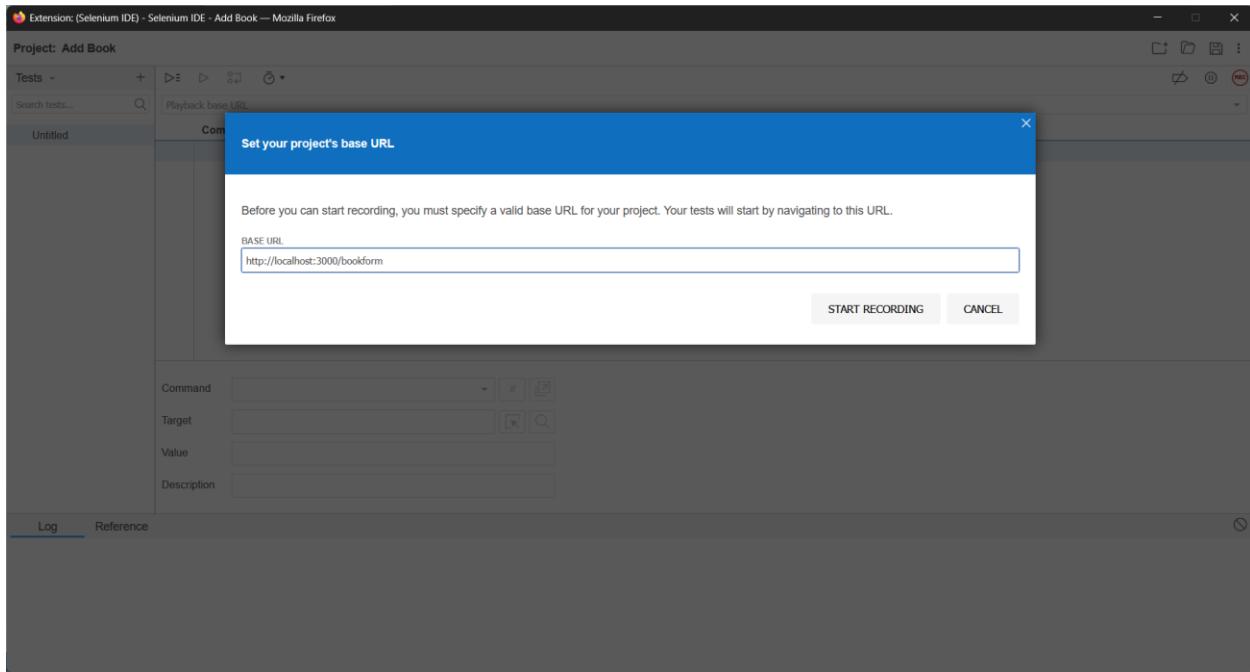
3. Project issues

The screenshot shows the SonarQube 'Issues' page, specifically for the 'LibraryManagementSystem' project. The 'Issues' tab is selected. A message at the top indicates that an embedded database is being used for evaluation purposes only. The main area lists several open bugs and code smells, each with a checkbox, an intentionality level (Maintainability), and a timestamp. The filters on the left allow searching by issue type (Bugs, Vulnerabilities, or Code Smells) and severity (Blocker, High, Medium, Low, Info). The filters also include Software Quality (Security, Reliability, Maintainability) and Clean Code Attribute (Consistency). The bottom of the page includes footer information about SonarQube technology and a build ID.

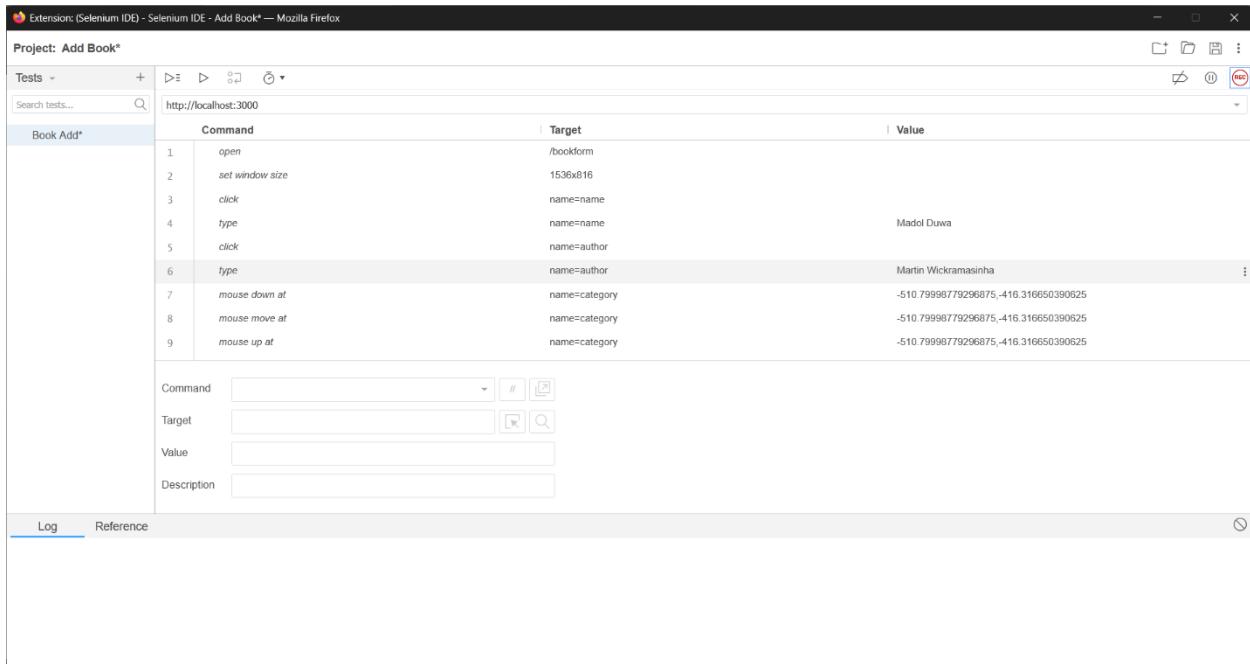
Selenium

Add Book

1. Selenium IDE with the base URL



2. Automation test script



The screenshot shows the Selenium IDE interface with a test project named "Add Book". The test case "Book Add*" is displayed, containing 16 commands. The last command, "assert alert", has a value of "Book added successfully!". Below the table, there are input fields for Command, Target, Value, and Description, and tabs for Log and Reference.

Command	Target	Value
9 mouse up at	name=category	-510.79998779296875,-416.316650390625
10 click	name=category	
11 select	name=category	label=Adventure
12 click	css=option:nth-child(3)	
13 click	name=publishedDate	
14 type	name=publishedDate	2024-01-17
15 click	css=submit.btn	
16 assert alert	Book added successfully!	

3. Selenium IDE after running the test case

The screenshot shows the Selenium IDE interface after running the "Book Add" test case. The test log at the bottom indicates that all 16 steps were successful ("OK"). The last step, "assertAlert", was also successful. The log entries show the timestamp for each step.

Step	Action	Status	Timestamp
1	✓ open	OK	18:49:56
2	✓ set window size	OK	18:49:57
3	✓ click	OK	18:49:57
4	✓ type	OK	18:49:57
5	✓ click	OK	18:49:57
6	✓ type	OK	18:49:57
7	✓ mouse down at	OK	18:49:57
8	✓ mouse move at	OK	18:49:57
9	✓ mouse up at	OK	18:49:57
11	select on name=category with value label=Adventure	OK	18:49:57
12	click on css=option:nth-child(3)	OK	18:49:57
13	click on name=publishedDate	OK	18:49:57
14	type on name=publishedDate with value 2024-01-17	OK	18:49:57
15	click on css=submit.btn	OK	18:49:57
16	assertAlert on Book added successfully!	OK	18:49:57
'Book Add' completed successfully			

Extension: (Selenium IDE) · Selenium IDE - Add Book* — Mozilla Firefox

Project: Add Book*

Tests + D: Search tests...

http://localhost:3000

	Command	Target	Value
9	✓ mouse up at	name=category	-510.79998779296875,-416.316650390625
10	✓ click	name=category	
11	✓ select	name=category	
12	✓ click	css=option:nth-child(3)	
13	✓ click	name=publishedDate	
14	✓ type	name=publishedDate	2024-01-17
15	✓ click	css=submit-btn	
16	✓ assert alert	Book added successfully!	

Command #

Target #

Value

Description

Log Reference

- 11. select on name=category with value label=Adventure OK 18:49:56
- 12. click on css=option:nth-child(3) OK 18:49:57
- 13. click on name=publishedDate OK 18:49:57
- 14. type on name=publishedDate with value 2024-01-17 OK 18:49:57
- 15. click on css=submit-btn OK 18:49:57
- 16. assertAlert on Book added successfully! OK 18:49:57

'Book Add' completed successfully 18:49:58

4. Verify items on the page

Extension: (Selenium IDE) · Selenium IDE - Add Book* — Mozilla Firefox

Project: Add Book*

Tests + D: Run current test Ctrl+R Search tests...

✓ Book Add*

	Command	Target	Value
1	✓ open	/bookform	
2	✓ set window size	1536x816	
3	✓ click	name=name	
4	✓ type	name=name	Madol Duwa
5	✓ click	name=author	
6	✓ type	name=author	Martin Wickramasingha
7	✓ mouse down at	name=category	-510.79998779296875,-416.316650390625
8	✓ mouse move at	name=category	-510.79998779296875,-416.316650390625
9	✓ mouse up at	name=category	-510.79998779296875,-416.316650390625

Command #

Target #

Value

Description

Log Reference

- 11. select on name=category with value label=Adventure OK 18:49:56
- 12. click on css=option:nth-child(3) OK 18:49:57
- 13. click on name=publishedDate OK 18:49:57
- 14. type on name=publishedDate with value 2024-01-17 OK 18:49:57
- 15. click on css=submit-btn OK 18:49:57
- 16. assertAlert on Book added successfully! OK 18:49:57

'Book Add' completed successfully 18:49:58

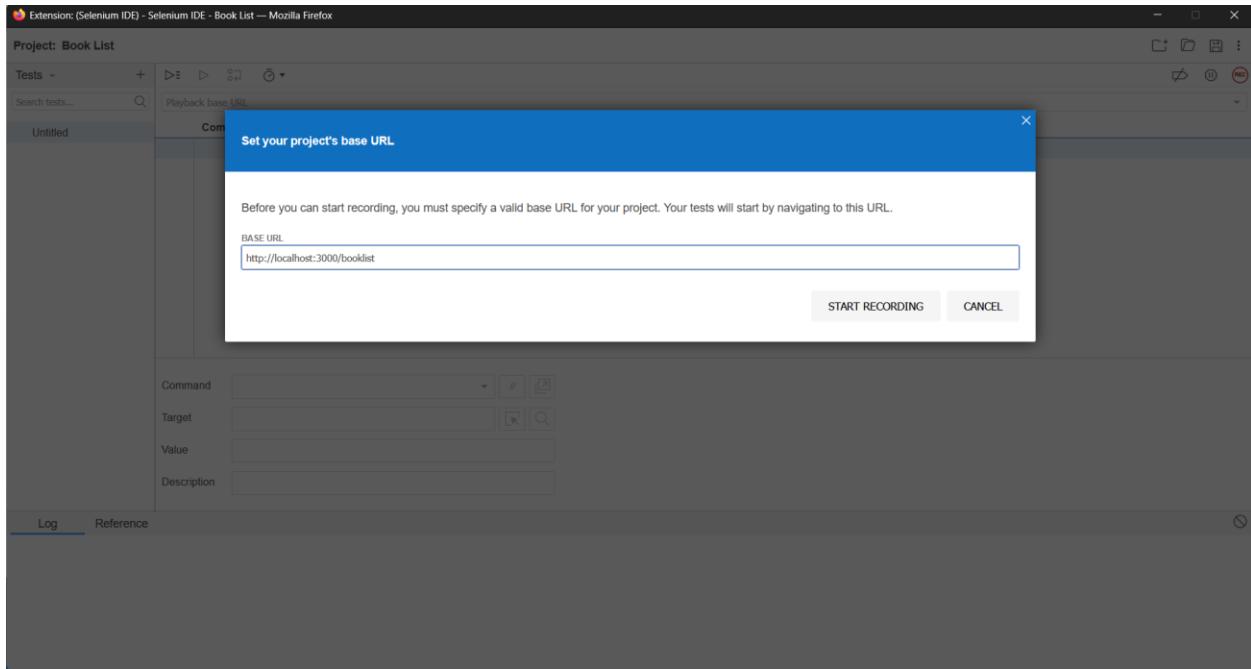
Extension: (Selenium IDE) - Selenium IDE - Add Book* — Mozilla Firefox

Project: Add Book*

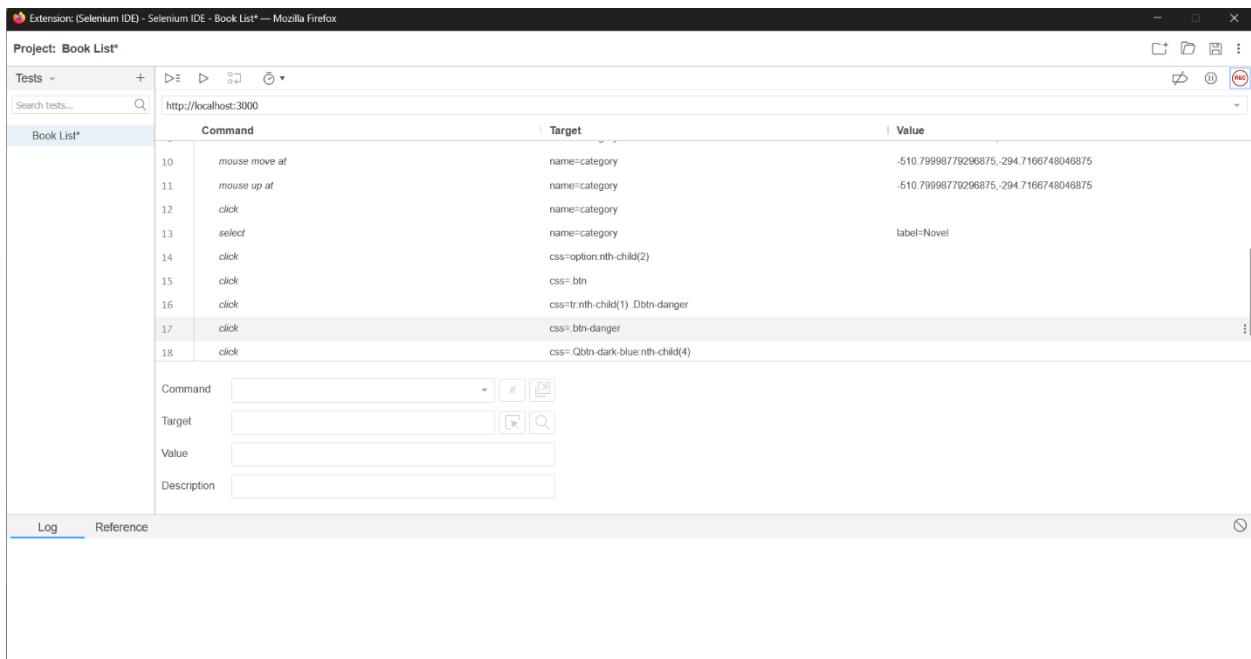
Tests	+	D	Run	Open	Close																											
Search tests...	<input type="text"/>																															
http://localhost:3000																																
<table border="1"> <thead> <tr> <th>Command</th> <th>Target</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>9 ✓ mouse up at</td> <td>name=category</td> <td>-510.7998779296875,-416.316650390625</td> </tr> <tr> <td>10 ✓ click</td> <td>name=category</td> <td></td> </tr> <tr> <td>11 ✓ select</td> <td>name=category</td> <td>label=Adventure</td> </tr> <tr> <td>12 ✓ click</td> <td>css=option:nth-child(3)</td> <td></td> </tr> <tr> <td>13 ✓ click</td> <td>name=publishedDate</td> <td></td> </tr> <tr> <td>14 ✓ type</td> <td>name=publishedDate</td> <td>2024-01-17</td> </tr> <tr> <td>15 ✓ click</td> <td>css=submit-btn</td> <td></td> </tr> <tr> <td>16 ✓ assert alert</td> <td>Book added successfully!</td> <td></td> </tr> </tbody> </table>						Command	Target	Value	9 ✓ mouse up at	name=category	-510.7998779296875,-416.316650390625	10 ✓ click	name=category		11 ✓ select	name=category	label=Adventure	12 ✓ click	css=option:nth-child(3)		13 ✓ click	name=publishedDate		14 ✓ type	name=publishedDate	2024-01-17	15 ✓ click	css=submit-btn		16 ✓ assert alert	Book added successfully!	
Command	Target	Value																														
9 ✓ mouse up at	name=category	-510.7998779296875,-416.316650390625																														
10 ✓ click	name=category																															
11 ✓ select	name=category	label=Adventure																														
12 ✓ click	css=option:nth-child(3)																															
13 ✓ click	name=publishedDate																															
14 ✓ type	name=publishedDate	2024-01-17																														
15 ✓ click	css=submit-btn																															
16 ✓ assert alert	Book added successfully!																															
<table border="1"> <tr> <td>Command</td> <td><input type="text"/></td> <td><input type="button"/></td> <td><input type="button"/></td> </tr> <tr> <td>Target</td> <td><input type="text"/></td> <td><input type="button"/></td> <td><input type="button"/></td> </tr> <tr> <td>Value</td> <td><input type="text"/></td> <td></td> <td></td> </tr> <tr> <td>Description</td> <td colspan="3"><input type="text"/></td> </tr> </table>						Command	<input type="text"/>	<input type="button"/>	<input type="button"/>	Target	<input type="text"/>	<input type="button"/>	<input type="button"/>	Value	<input type="text"/>			Description	<input type="text"/>													
Command	<input type="text"/>	<input type="button"/>	<input type="button"/>																													
Target	<input type="text"/>	<input type="button"/>	<input type="button"/>																													
Value	<input type="text"/>																															
Description	<input type="text"/>																															
<p>Log Reference</p> <p>11. select on name=category with value label=Adventure OK 12. click on css=option:nth-child(3) OK 13. click on name=publishedDate OK 14. type on name=publishedDate with value 2024-01-17 OK 15. click on css=submit-btn OK 16. assertAlert on Book added successfully! OK 'Book Add' completed successfully</p>																																

Update Book

1. Selenium IDE with the base URL



2. Automation test script



Project: Book List*

Tests + DΞ D o G

Search tests... Q

<http://localhost:3000>

	Command	Target	Value
1	open	/booklist	
2	set window size	1536x816	
3	click	css=form-control	
4	type	css=form-control	2025042955
5	click	css=container	
6	click	css=Qbtn-dark-blue:nth-child(1)	
7	click	css=btn	
8	click	css=Ebtn-light-blue	
9	mouse down at	name=category	-510.79998779296875,-294.7166748046875

Command R [C]

Target T S

Value

Description

Log Reference Q

3. Selenium IDE after running the test case

Project: Book List*

Executing ✓ Book List*

<http://localhost:3000>

	Command	Target	Value
1	✓ open	/booklist	
2	✓ set window size	1536x816	
3	✓ click	css=form-control	
4	✓ type	css=form-control	2025042955
5	✓ click	css=container	
6	✓ click	css=Ebtn-light-blue	
7	✓ mouse down at	name=category	-510.79998779296875,-294.7166748046875
8	✓ mouse move at	name=category	-510.79998779296875,-294.7166748046875
9	✓ mouse up at	name=category	-510.79998779296875,-294.7166748046875

Command R [C]

Target T S

Value

Description

Runs: 1 Failures: 0 Log Reference Q

```

1. open on /booklist OK
2. setWindowSize on 1536x816 OK
3. click on css= form-control OK
4. type on css= form-control with value 2025042955 OK
5. click on css= container OK
    
```

18:56:46 18:56:46 18:56:46 18:56:46 18:56:46

Extension: (Selenium IDE) - Selenium IDE - Book List* — Mozilla Firefox

Project: Book List*

Executing ✓ Book List* http://localhost:3000

Command	Target	Value
✓ mouse up at	name=category	-510.79998779296875,-294.7166748046875
✓ click	name=category	
✓ select	name=category	
✓ click	css=option:nth-child(2)	
✓ click	css=bln	
✓ click	css=tr:nth-child(1).Dbtn-danger	
✓ click	css=btn-danger	
✓ click	css=Qbtn-dark-blue nth-child(4)	

Run: 1 Failures: 0

Log Reference

```
0. mouseUpAt on name=category with value -510.79998779296875,-294.7166748046875 OK
1. click on name=category OK
2. select on name=category with value label=Novel OK
3. click on css=option nth-child(2) OK
```

4. Verify items on the page

Extension: (Selenium IDE) - Selenium IDE - Book List* — Mozilla Firefox

Project: Book List*

Executing ✓ Book List* http://localhost:3000

Command	Target	Value
✓ open	/booklist	
✓ set window size	1536x816	
✓ click	css=form-control	
✓ type	css=form-control	2025042955
✓ click	css=container	
✓ click	css=Ebtn-light-blue	
✓ mouse down at	name=category	-510.79998779296875,-294.7166748046875
✓ mouse move at	name=category	-510.79998779296875,-294.7166748046875
✓ mouse up at	name=category	-510.79998779296875,-294.7166748046875

Run: 1 Failures: 0

Log Reference

```
1. open on /booklist OK
2. setWindowSize on 1536x816 OK
3. click on css= form-control OK
4. type on css= form-control with value 2025042955 OK
5. click on css= container OK
```

Extension: (Selenium IDE) - Selenium IDE - Book List* — Mozilla Firefox

Project: Book List*

Executing ▾

✓ Book List*

http://localhost:3000

	Command	Target	Value
9	✓ mouse up at	name=category	-510.79990779290073,-294.7166748046875
10	✓ click	name=category	
11	✓ select	name=category	
12	✓ click	css=option:nth-child(2)	
13	✓ click	css=btn	
14	✓ click	css=tr:nth-child(1) .btn-danger	
15	✓ click	css=btn-danger	
16	✓ click	css=.Qbtn-dark-blue nth-child(4)	

Command

Target

Value

Description

Runs: 1 Failures: 0

Log Reference

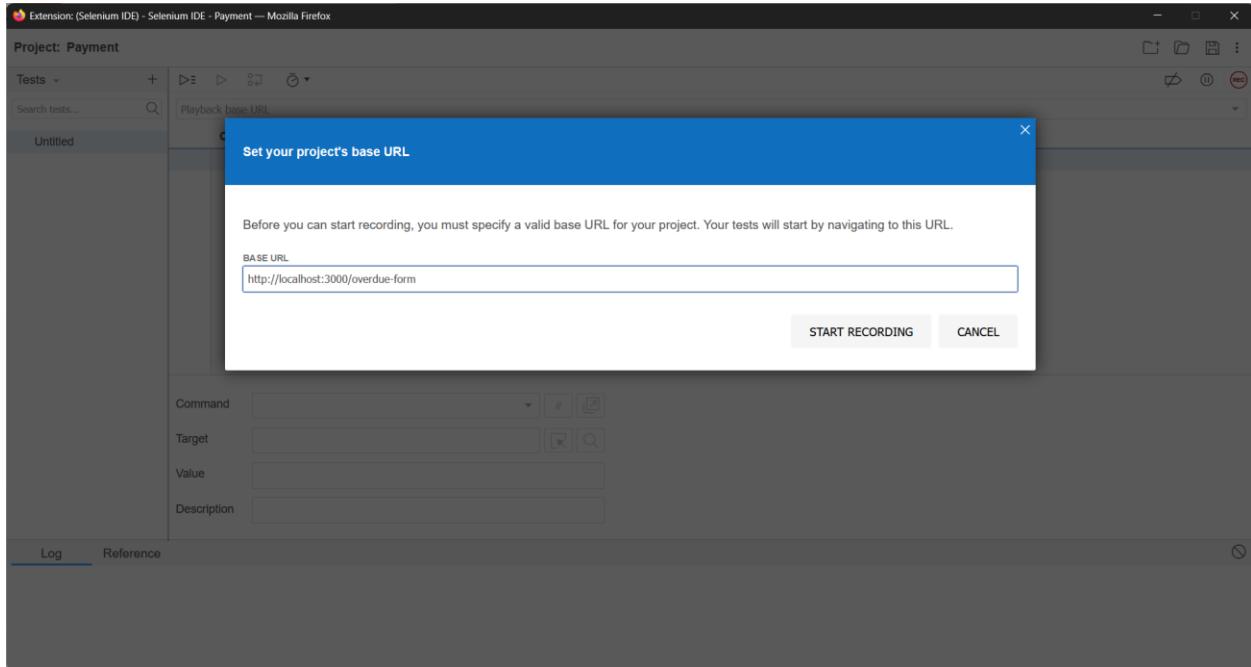
```

o mouseUpAt on name=category with value -510.79990779290073,-294.7166748046875 OK
9. mouseUpAt on name=category with value -510.79990779290073,-294.7166748046875 OK
10. click on name=category OK
11. select on name=category with value label=Novel OK
12. click on css=option:nth-child(2) OK

```

Over Due Form

1. Selenium IDE with the base URL



2. Automation test script

The screenshot shows the Selenium IDE interface in Mozilla Firefox. The main window displays a test script titled "Payment*". The test steps are listed in a table:

	Command	Target	Value
1	open	/overdue-form	
2	set window size	1536x816	
3	click	name=ISBN	
4	type	name=ISBN	12345
5	click	name=BookName	
6	type	name=BookName	Madol Duwa
7	click	name=BorrowerName	
8	type	name=BorrowerName	Anuk
9	click	name=BorrowDate	

Below the table, there are input fields for "Command", "Target", "Value", and "Description". The "Log" tab is selected at the bottom of the interface.

Extension: (Selenium IDE) - Selenium IDE - Payment* — Mozilla Firefox

Project: Payment*

Test	+	▶	▶	⋮
Search tests...	Q			
http://localhost:3000				
Payment*		Command	Target	Value
10	<i>type</i>	name=BorrowDate	2025-04-11	
11	<i>click</i>	name=ReturnDate		
12	<i>type</i>	name=ReturnDate	2025-04-23	
13	<i>click</i>	name=OverdueDays		
14	<i>click</i>	css=form-group:nth-child(7) > input		
15	<i>click</i>	css=custom-btn-submit		
16	<i>click</i>	css=td:nth-child(5)		
17	<i>click</i>	css=td:nth-child(5)		
18	<i>double click</i>	css=td:nth-child(5)		

Command

Target

Value

Description

Log Reference

Extension: (Selenium IDE) - Selenium IDE - Payment* — Mozilla Firefox

Project: Payment*

Test	+	▶	▶	⋮
Search tests...	Q			
http://localhost:3000				
Payment*		Command	Target	Value
18	<i>double click</i>	css=td:nth-child(5)		
19	<i>click</i>	css=.container		
20	<i>click</i>	css=.search-input		
21	<i>click</i>	css=btn-success		
22	<i>click</i>	css=btn-primary		
23	<i>click</i>	css=input:nth-child(2)		
24	<i>type</i>	css=input:nth-child(2)	C:\fakepath\n8.1.png	
25	<i>click</i>	css=btn		
26	<i>click</i>	css=btn-success		

Command

Target

Value

Description

Log Reference

Extension: Selenium IDE - Selenium IDE - Payment* — Mozilla Firefox

Project: Payment*

Tests +     

Search tests... 

http://localhost:3000

	Command	Target	Value
27	click	css=.btn-secondary	
28	click	css=.input-section:nth-child(3) > input	
29	type	css=.input-section:nth-child(3) > input	123456789456
30	click	css=.input-section:nth-child(4) > input	
31	type	css=.input-section:nth-child(4) > input	365
32	click	css=.otp-section > input	
33	type	css=.otp-section > input	123456
34	click	css=.btn-primary	
35	assert alert	OTP has been sent to your registered mobile number.	

Command   

Target   

Value

Description

Log  Reference 

Extension: Selenium IDE - Selenium IDE - Payment* — Mozilla Firefox

Project: Payment*

Tests +     

Search tests... 

http://localhost:3000

	Command	Target	Value
31	type	css=.input-section:nth-child(4) > input	365
32	click	css=.otp-section > input	
33	type	css=.otp-section > input	123456
34	click	css=.btn-primary	
35	assert alert	OTP has been sent to your registered mobile number.	
36	click	css=.btn-success	
37	assert alert	Payment successful for ISBN: undefined	

Command   

Target   

Value

Description

Log  Reference 

3. Selenium IDE after running the test case

Extension: (Selenium IDE) - Selenium IDE - Payment* — Mozilla Firefox

Project: Payment*

Executing ▾ http://localhost:3000

Command	Target	Value
1 ✓ open	/overdue-form	
2 ✓ set window size	1536x816	
3 ✓ click	name=ISBN	
4 ✓ type	name=ISBN	12345
5 ✓ click	name=BookName	
6 ✓ type	name=BookName	Madol Duwa
7 ✓ click	name=BorrowerName	
8 ✓ type	name=BorrowerName	Anuk
9 ✓ click	name=BorrowDate	

Command: assert alert

Target: Payment successful for ISBN: undefined

Value:

Description:

Runs: 1 Failures: 0

Log Reference

- 1. open on /overdue-form OK 19:10:23
- 2. setWindowSize on 1536x816 OK 19:10:23
- 3. click on name=ISBN OK 19:10:24
- 4. type on name=ISBN with value 12345 OK 19:10:24
- 5. click on name=BookName OK 19:10:24

Extension: (Selenium IDE) - Selenium IDE - Payment* — Mozilla Firefox

Project: Payment*

Executing ▾ http://localhost:3000

Command	Target	Value
10 ✓ type	name=BorrowDate	2025-04-11
11 ✓ click	name=ReturnDate	
12 ✓ type	name=ReturnDate	2025-04-23
13 ✓ click	name=OverdueDays	
14 ✓ click	css=.form-group:nth-child(7) > input	
15 ✓ click	css=custom-btn-submit	
16 ✓ click	css=td:nth-child(5)	
17 ✓ click	css=td:nth-child(5)	
18 ✓ double click	css=td:nth-child(5)	

Command: assert alert

Target: Payment successful for ISBN: undefined

Value:

Description:

Runs: 1 Failures: 0

Log Reference

- 1. open on /overdue-form OK 19:10:23
- 2. setWindowSize on 1536x816 OK 19:10:23
- 3. click on name=ISBN OK 19:10:24
- 4. type on name=ISBN with value 12345 OK 19:10:24
- 5. click on name=BookName OK 19:10:24

Extension: (Selenium IDE) - Selenium IDE - Payment* — Mozilla Firefox

Project: Payment*

Executing ▾

✓ Payment*

http://localhost:3000

Command	Target	Value
18 ✓ double click	css=id:nth-child(5)	
19 ✓ click	css= container	
20 ✓ click	css=.search-input	
21 ✓ click	css=.btn-success	
22 ✓ click	css=.btn-primary	
23 ✓ click	css=input:nth-child(2)	
24 ✓ click	css= bfn	
25 ✓ click	css= btn-success	
26 ✓ click	css= input_section:nth-child(3)> input	

Command: assert alert

Target: Payment successful for ISBN: undefined

Description:

Runs: 1 Failures: 0

Log Reference

```
18. doubleclick on css=id:nth-child(5) OK
19. click on css= container OK
20. click on css=.search-input OK
21. click on css=.btn-success OK
22. click on css=.btn-primary OK
```

19.10.26 19.10.26 19.10.26 19.10.27

Extension: (Selenium IDE) - Selenium IDE - Payment* — Mozilla Firefox

Project: Payment*

Executing ▾

✓ Payment*

http://localhost:3000

Command	Target	Value
27 ✓ type	css= input_section:nth-child(3)> input	123456789456
28 ✓ click	css= input_section:nth-child(4)> input	
29 ✓ type	css= input_section:nth-child(4)> input	365
30 ✓ click	css= otp-section > input	
31 ✓ type	css= otp-section > input	123456
32 ✓ click	css= btn-primary	
33 ✓ assert alert	OTP has been sent to your registered mobile number.	
34 ✓ click	css= btn-success	
35 ✓ assert alert	Payment successful for ISBN: undefined	

Command: assert alert

Target: Payment successful for ISBN: undefined

Description:

Runs: 1 Failures: 0

Log Reference

```
31. type on css= otp-section > input with value 123456 OK
32. click on css= btn-primary OK
33. assertAlert on OTP has been sent to your registered mobile number. OK
34. click on css= btn-success OK
35. assertAlert on Payment successful for ISBN: undefined OK
```

19.10.38 19.10.38 19.10.38 19.10.39 19.10.39

4. Verify items on the page

Selenium IDE - Selenium IDE - Payment* — Mozilla Firefox

Project: Payment*

Executing: ✓ Payment*

http://localhost:3000

Command	Target	Value
1 ✓ open	/overdue-form	
2 ✓ set window size	1536x816	
3 ✓ click	name=ISBN	
4 ✓ type	name=ISBN	12345
5 ✓ click	name=BookName	
6 ✓ type	name=BookName	Madol Duwa
7 ✓ click	name=BorrowerName	
8 ✓ type	name=BorrowerName	Anuk
9 ✓ click	name=BorrowDate	

Command: assert alert

Target: Payment successful for ISBN: undefined

Value:

Description:

Runs: 1 Failures: 0

Log Reference

- 1. open on /overdue-form OK 19:10:23
- 2. setWindowSize on 1536x816 OK 19:10:23
- 3. click on name=ISBN OK 19:10:24
- 4. type on name=ISBN with value 12345 OK 19:10:24
- 5. click on name=BookName OK 19:10:24

Selenium IDE - Selenium IDE - Payment* — Mozilla Firefox

Project: Payment*

Executing: ✓ Payment*

http://localhost:3000

Command	Target	Value
10 ✓ type	name=BorrowDate	2025-04-11
11 ✓ click	name=ReturnDate	
12 ✓ type	name=ReturnDate	2025-04-23
13 ✓ click	name=OverdueDays	
14 ✓ click	css=.form-group:nth-child(7) > input	
15 ✓ click	css=custom-btn-submit	
16 ✓ click	css=td:nth-child(6)	
17 ✓ click	css=td:nth-child(5)	
18 ✓ double click	css=td:nth-child(5)	

Command: assert alert

Target: Payment successful for ISBN: undefined

Value:

Description:

Runs: 1 Failures: 0

Log Reference

- 1. open on /overdue-form OK 19:10:23
- 2. setWindowSize on 1536x816 OK 19:10:23
- 3. click on name=ISBN OK 19:10:24
- 4. type on name=ISBN with value 12345 OK 19:10:24
- 5. click on name=BookName OK 19:10:24

Extension: (Selenium IDE) - Selenium IDE - Payment* — Mozilla Firefox

Project: Payment*

Executing ▾

✓ Payment*

http://localhost:3000

Command	Target	Value
18 ✓ double click	css=id nth-child(5)	
19 ✓ click	css=container	
20 ✓ click	css=search-input	
21 ✓ click	css=btn-success	
22 ✓ click	css=btn-primary	
23 ✓ click	css=input:nth-child(2)	
24 ✓ click	css=btn	
25 ✓ click	css=btn-success	
26 ✓ click	css=input-section:nth-child(3) > input	

Command: assert alert

Target: Payment successful for ISBN: undefined

Value:

Description:

Runs: 1 Failures: 0

Log Reference

```
18: double click on css=id nth-child(5) OK
19: click on css=container OK
20: click on css=search-input OK
21: click on css=btn-success OK
22: click on css=btn-primary OK
23: click on css=input:nth-child(2) OK
24: click on css=btn OK
25: click on css=btn-success OK
26: click on css=input-section:nth-child(3) > input OK
```

19.10.26 19.10.26 19.10.26 19.10.27

Extension: (Selenium IDE) - Selenium IDE - Payment* — Mozilla Firefox

Project: Payment*

Executing ▾

✓ Payment*

http://localhost:3000

Command	Target	Value
27 ✓ type	css=input-section:nth-child(3) > input	123456789456
28 ✓ click	css=input-section:nth-child(4) > input	
29 ✓ type	css=input-section:nth-child(4) > input	365
30 ✓ click	css=otp-section > input	
31 ✓ type	css=otp-section > input	123456
32 ✓ click	css=btn-primary	
33 ✓ assert alert	OTP has been sent to your registered mobile number.	
34 ✓ click	css=btn-success	
35 ✓ assert alert	Payment successful for ISBN: undefined	

Command: assert alert

Target: Payment successful for ISBN: undefined

Value:

Description:

Runs: 1 Failures: 0

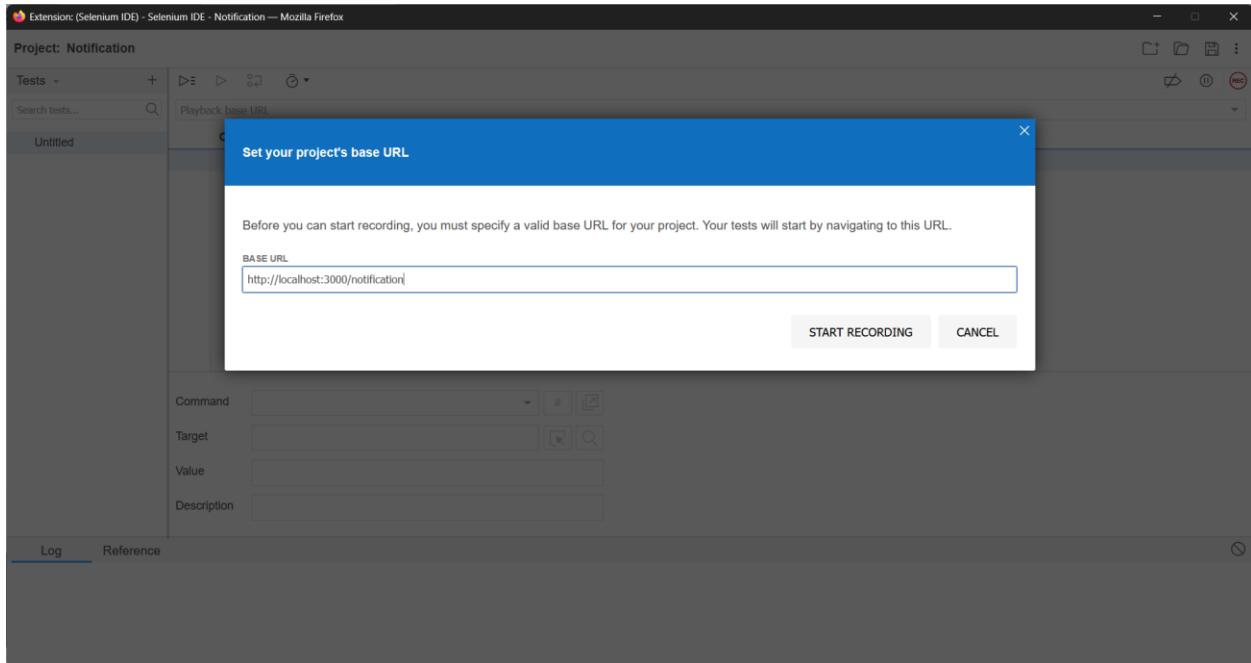
Log Reference

```
31: type on css=otp-section > input with value 123456 OK
32: click on css=btn-primary OK
33: assertAlert on OTP has been sent to your registered mobile number. OK
34: click on css=btn-success OK
35: assertAlert on Payment successful for ISBN: undefined OK
```

19.10.38 19.10.38 19.10.38 19.10.39 19.10.39

Notification

1. Selenium IDE with the base URL



2. Automation test script

The screenshot shows the Selenium IDE interface with a test script named "Notification". The test script table has three columns: Command, Target, and Value. The steps are as follows:

Command	Target	Value
open	/notification	
set window size	1536x816	
click	name=message	
type	name=message	you have 3 points
click	css=label:nth-child(2) > input	
click	css=.radio-group > label:nth-child(1)	
click	css=label:nth-child(2) > input	
click	css=button	
assert alert		Notification Submitted Successfully!

Below the table, there are fields for "Command", "Target", "Value", and "Description". The "Log" tab is selected at the bottom left.

The screenshot shows the Selenium IDE extension in Mozilla Firefox. The title bar reads "Extension: (Selenium IDE) - Selenium IDE - Notification* — Mozilla Firefox". The main interface is titled "Project: Notification*". The "Tests" dropdown is open, and the URL "http://localhost:3000" is entered in the search bar. The test log table contains the following entries:

	Command	Target	Value
10	click	css=.edit-btn	
11	click	name=message	
12	type	name=message	
13	click	css=button:nth-child(9)	you have 8 points
14	assert alert	Notification Submitted Successfully!	
15	click	css=delete-btn	
16	assert alert	Notification Deleted Successfully!	

Below the table are input fields for "Command", "Target", "Value", and "Description". At the bottom, there are tabs for "Log" (which is selected) and "Reference".

3. Selenium IDE after running the test case

The screenshot shows the Selenium IDE extension in Mozilla Firefox after the test case has been run. The title bar and project name are the same as the previous screenshot. The test log table now shows green checkmarks next to each command, indicating successful execution. The log table entries are identical to the ones in the first screenshot.

	Command	Target	Value
1	✓ open	/notification	
2	✓ set window size	1536x816	
3	✓ click	name=message	
4	✓ type	name=message	you have 3 points
5	✓ click	css=label:nth-child(2) > input	
6	✓ click	css=.radio-group > label:nth-child(1)	
7	✓ click	css=label:nth-child(2) > input	
8	✓ click	css=button	

Below the table are input fields for "Command", "Target", "Value", and "Description". At the bottom, there are tabs for "Log" (selected) and "Reference".

Running 'Notification'

```

1. open on /notification OK
2. setWindowSize on 1536x816 OK
3. click on name=message OK
4. type on name=message with value you have 3 points OK
    
```

Timestamps for the log entries are shown at the right end of each line.

4. Verify items on the page

The screenshot shows the Selenium IDE extension in Mozilla Firefox. The title bar reads "Extension: (Selenium IDE) - Selenium IDE - Notification* - Mozilla Firefox". The main interface displays a test log for a "Notification" project. The log contains the following steps:

Step	Command	Target	Value
1	✓ open	/notification	
2	✓ set window size	1536x816	
3	✓ click	name=message	
4	✓ type	name=message	you have 3 points
5	✓ click	css=label:nth-child(2) > input	
6	✓ click	css=radio-group > label:nth-child(1)	
7	✓ click	css=label:nth-child(2) > input	
8	✓ click	css=button	

Below the log, there are four input fields: "Command", "Target", "Value", and "Description", each with a dropdown arrow and a search icon.

Extension: (Selenium IDE) - Selenium IDE - Notification* — Mozilla Firefox

Project: Notification*

Tests + [undo] [redo] [refresh] [rec]

Search tests... [x]

http://localhost:3000

Command	Target	Value
9 ✓ assert alert	Notification Submitted Successfully!	
10 ✓ click	css=edit-btn	
11 ✓ click	name=message	
12 ✓ type	name=message	
13 ✓ click	css=button:nth-child(9)	you have 8 points
14 ✓ assert alert	Notification Submitted Successfully!	
15 ✓ click	css=delete-btn	
16 ✓ assert alert	Notification Deleted Successfully!	

Command [New] [Edit]

Target [New] [Edit]

Value [New] [Edit]

Description [New] [Edit]

Log **Reference**

```

12. type on name=message with value you have 8 points OK
13. click on css=button:nth-child(9) OK
14. assertAlert on Notification Submitted Successfully! OK
15. click on css= delete-btn OK
16. assertAlert on Notification Deleted Successfully! OK
'Notification' completed successfully

```

[Close]