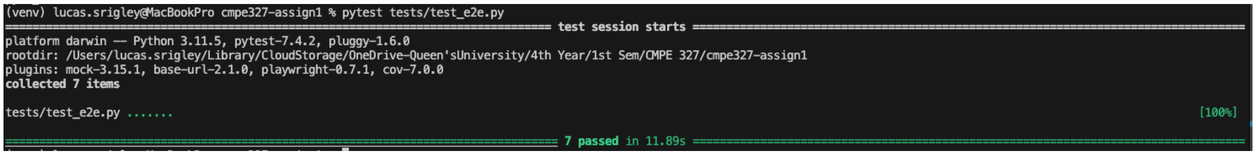
CMPE 327 Assignment 4

1. Student Information

Name: Lucas Srigley
Student ID: 20289448
Date: November 18, 2025
GitHub URL: https://github.com/lucas-srigley/cisc327-library-management-a2-9448

2. E2E Testing Approach

I used Playwright for my E2E testing. The features I tested are adding, borrowing, searching, and returning for a book. I also tested the patron status, invalid ISBN, and navigation links. The different assertions I made were visibility of UI elements (buttons, text, etc), success/error feedback, and availability count changes. My tests passed are shown below:



```
(venv) lucas.srigley@MacBookPro cmpe327-assign1 % pytest tests/test_e2e.py
============================================================================ test session starts ============================================================================
platform darwin -- Python 3.11.5, pytest-7.4.2, pluggy-1.6.0
rootdir: /Users/lucas.srigley/Library/CloudStorage/OneDrive-Queen'sUniversity/4th Year/1st Sem/CMPE 327/cmpe327-assign1
plugins: mock-3.15.1, base-url-2.1.0, playwright-0.7.1, cov-7.0.0
collected 7 items

tests/test_e2e.py .......                                                                                                                                            [100%]

============================================================================= 7 passed in 11.89s =============================================================================
```

3. Execution Instructions

Tests

pytest tests/test_e2e.py

Docker Containerization

docker build -t library-app .
docker run -p 5000:5000 library-app

Docker Hub

docker login
docker tag library-app srigs97/library-app:v1
docker push srigs97/library-app:v1
docker rmi srigs97/library-app:v1
docker pull srigs97/library-app:v1
docker run -p 5000:5000 srigs97/library-app:v1

4. Test Case Summary

Test Case	**Actions**	**Expected Result**
Add book	Fill form with data for book and submit	Book is added to catalog and is shown in the table
Borrow book	Enter patron ID and submit	Success message, update availability
Search book	Type search term/search type	Search results are correct based off search

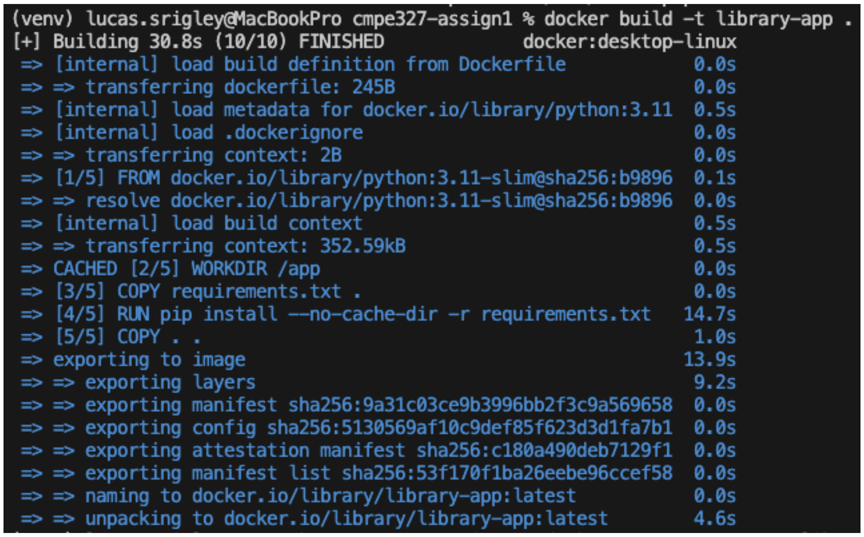
Return book	Borrow book and then return it by filling out patron ID/book ID and submit	Success message and update availability
Patron status	Enter patron ID and submit	Display patron table info including any borrowed books, late fees, and history
Invalid ISBN	Test adding a book with ISBN < 13 digits	Error message for invalid ISBN
Navigation links	Click on each navigation link	Successfully navigates to each page

5. Dockerization Process

Steps

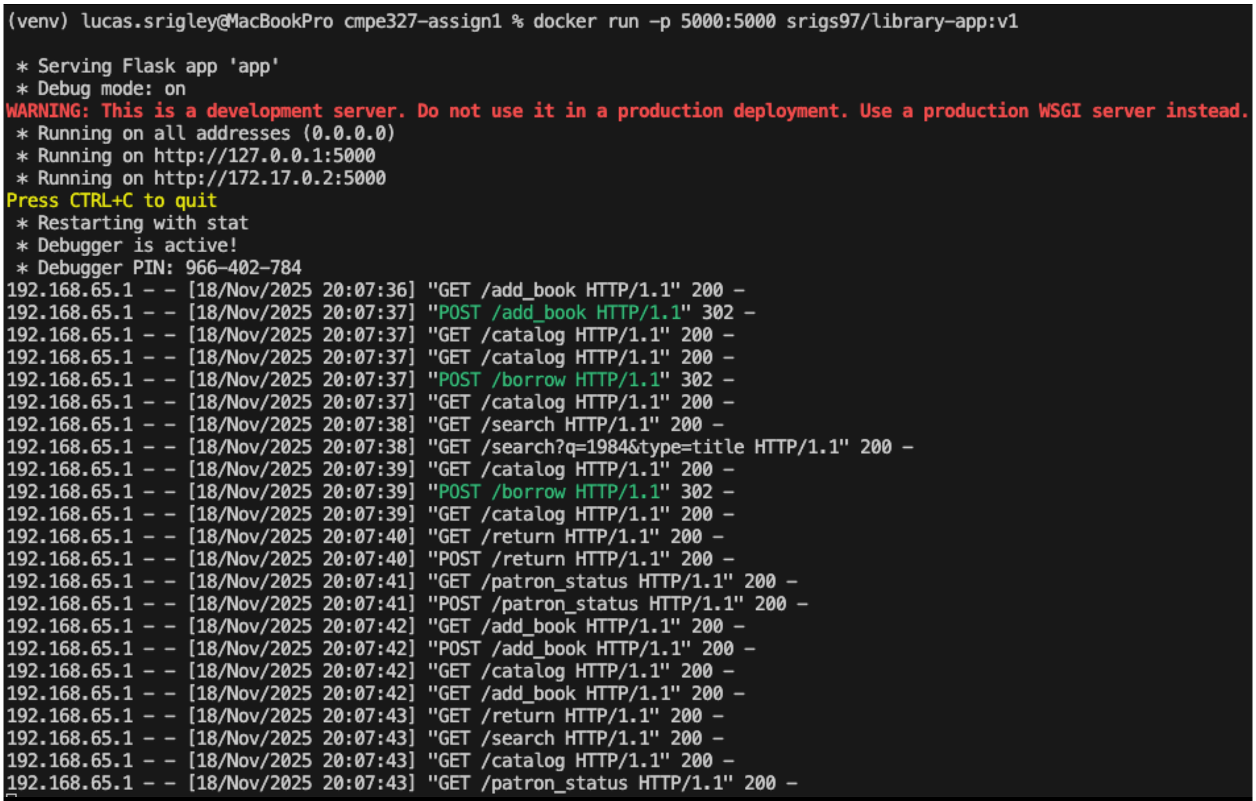
Created Dockerfile with Python 3.11-slim, copied requirements.txt file and used it to install dependencies, copied app source files, exposed port 5000, and set Flask entry point as app.py.

Build



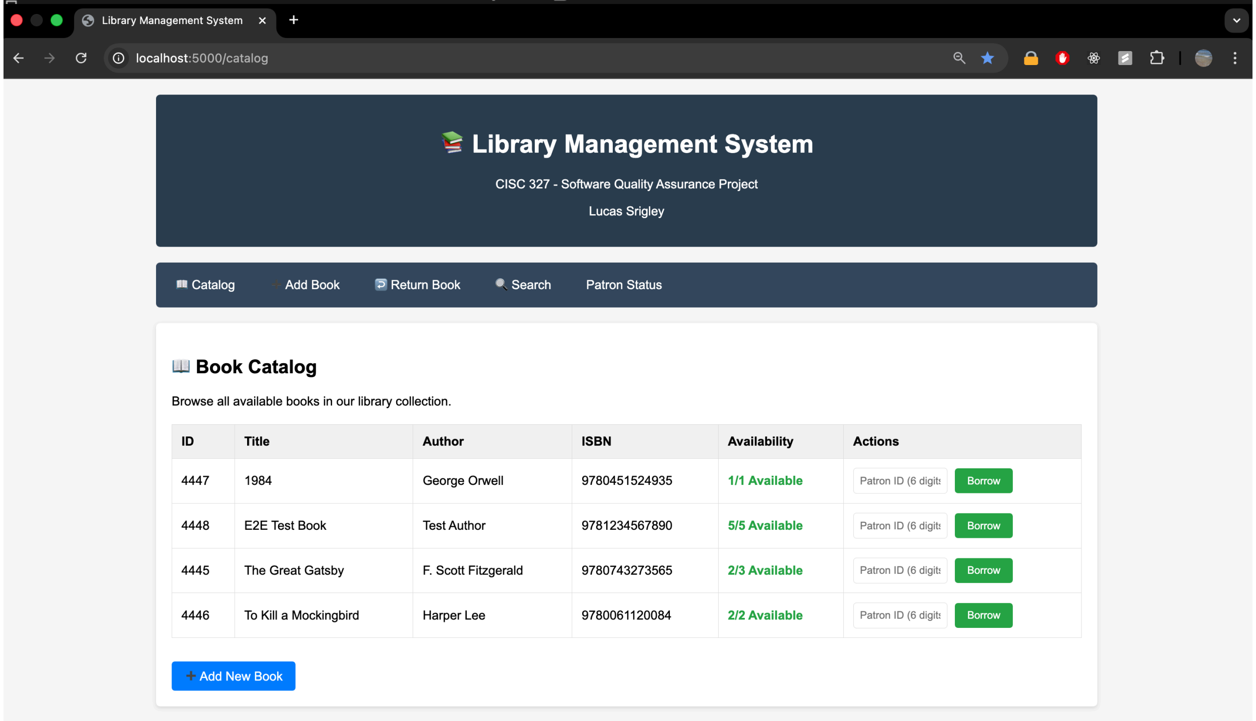
```
(venv) lucas.srigley@MacBookPro cmpe327-assign1 % docker build -t library-app .
[+] Building 30.8s (10/10) FINISHED                 docker:desktop-linux
 => [internal] load build definition from Dockerfile                0.0s
 => => transferring dockerfile: 245B                                0.0s
 => [internal] load metadata for docker.io/library/python:3.11      0.5s
 => [internal] load .dockerignore                                   0.0s
 => => transferring context: 2B                                     0.0s
 => [1/5] FROM docker.io/library/python:3.11-slim@sha256:b9896      0.1s
 => => resolve docker.io/library/python:3.11-slim@sha256:b9896      0.0s
 => [internal] load build context                                   0.5s
 => => transferring context: 352.59kB                               0.5s
 => CACHED [2/5] WORKDIR /app                                       0.0s
 => [3/5] COPY requirements.txt .                                   0.0s
 => [4/5] RUN pip install --no-cache-dir -r requirements.txt       14.7s
 => [5/5] COPY . .                                                  1.0s
 => exporting to image                                             13.9s
 => => exporting layers                                             9.2s
 => => exporting manifest sha256:9a31c03ce9b3996bb2f3c9a569658      0.0s
 => => exporting config sha256:5130569af10c9def85f623d3d1fa7b1      0.0s
 => => exporting attestation manifest sha256:c180a490deb7129f1      0.0s
 => => exporting manifest list sha256:53f170f1ba26eebe96ccef58      0.0s
 => => naming to docker.io/library/library-app:latest               0.0s
 => => unpacking to docker.io/library/library-app:latest            4.6s
```

Run



```
(venv) lucas.srigley@MacBookPro cmpe327-assign1 % docker run -p 5000:5000 srigs97/library-app:v1

 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.2:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 966-402-784
192.168.65.1 - - [18/Nov/2025 20:07:36] "GET /add_book HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:37] "POST /add_book HTTP/1.1" 302 -
192.168.65.1 - - [18/Nov/2025 20:07:37] "GET /catalog HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:37] "GET /catalog HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:37] "POST /borrow HTTP/1.1" 302 -
192.168.65.1 - - [18/Nov/2025 20:07:37] "GET /catalog HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:38] "GET /search HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:38] "GET /search?q=1984&type=title HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:39] "GET /catalog HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:39] "POST /borrow HTTP/1.1" 302 -
192.168.65.1 - - [18/Nov/2025 20:07:39] "GET /catalog HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:40] "GET /return HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:40] "POST /return HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:41] "GET /patron_status HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:41] "POST /patron_status HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:42] "GET /add_book HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:42] "POST /add_book HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:42] "GET /catalog HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:42] "GET /add_book HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:43] "GET /return HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:43] "GET /search HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:43] "GET /catalog HTTP/1.1" 200 -
192.168.65.1 - - [18/Nov/2025 20:07:43] "GET /patron_status HTTP/1.1" 200 -
```



📚 Library Management System

CISC 327 - Software Quality Assurance Project

Lucas Srigley

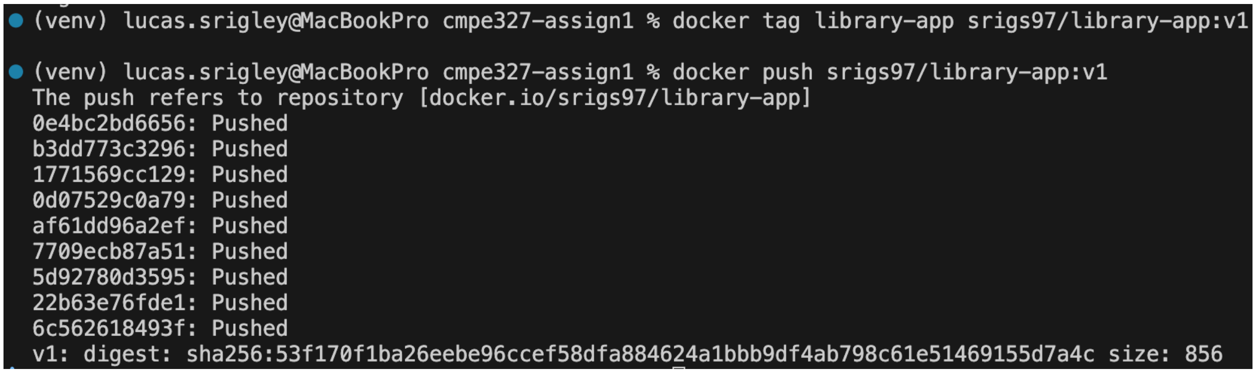
📖 Book Catalog

Browse all available books in our library collection.

ID	Title	Author	ISBN	Availability	Actions
4447	1984	George Orwell	9780451524935	1/1 Available	Patron ID (6 digit Borrow
4448	E2E Test Book	Test Author	9781234567890	5/5 Available	Patron ID (6 digit Borrow
4445	The Great Gatsby	F. Scott Fitzgerald	9780743273565	2/3 Available	Patron ID (6 digit Borrow
4446	To Kill a Mockingbird	Harper Lee	9780061120084	2/2 Available	Patron ID (6 digit Borrow

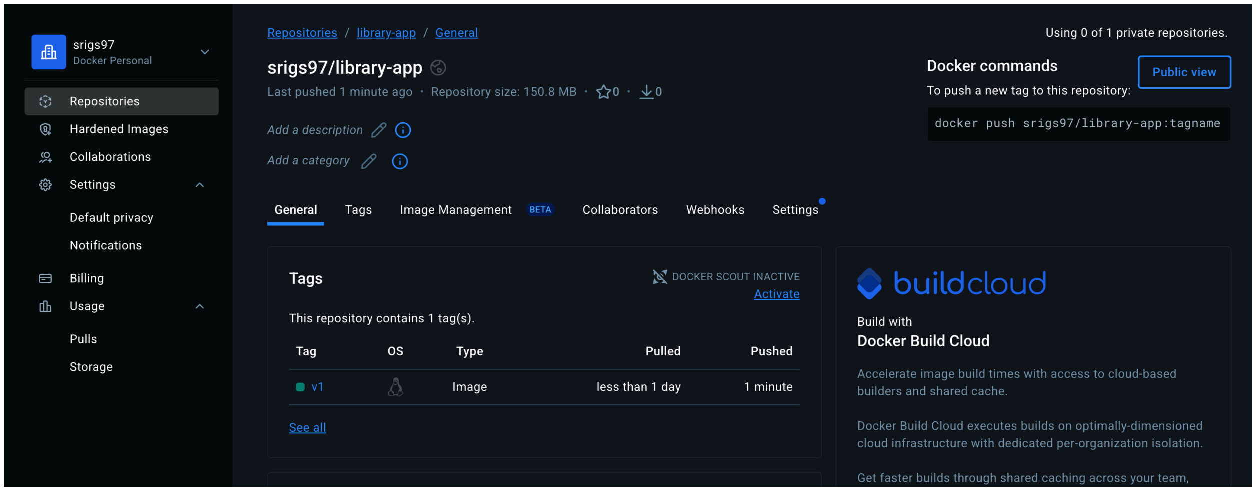
6. Docker Hub Deployment

Push



```
(venv) lucas.srigley@MacBookPro cmpe327-assign1 % docker tag library-app srigs97/library-app:v1

(venv) lucas.srigley@MacBookPro cmpe327-assign1 % docker push srigs97/library-app:v1
The push refers to repository [docker.io/srigs97/library-app]
0e4bc2bd6656: Pushed
b3dd773c3296: Pushed
1771569cc129: Pushed
0d07529c0a79: Pushed
af61dd96a2ef: Pushed
7709ecb87a51: Pushed
5d92780d3595: Pushed
22b63e76fde1: Pushed
6c562618493f: Pushed
v1: digest: sha256:53f170f1ba26eebe96ccef58dfa884624a1bbb9df4ab798c61e51469155d7a4c size: 856
```



Repositories / library-app / General

Using 0 of 1 private repositories.

srigs97/library-app

Last pushed 1 minute ago · Repository size: 150.8 MB · ☆0 · ⤓0

Add a description

Add a category

Docker commands

To push a new tag to this repository:

```
docker push srigs97/library-app:tagname
```

Tags

DOCKER SCOUT INACTIVE
Activate

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
v1	🐧	Image	less than 1 day	1 minute

See all

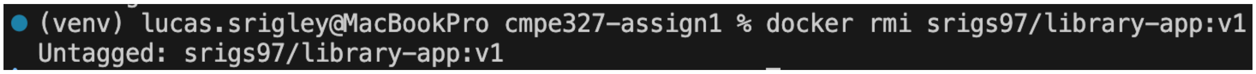
Build with
Docker Build Cloud

Accelerate image build times with access to cloud-based builders and shared cache.

Docker Build Cloud executes builds on optimally-dimensioned cloud infrastructure with dedicated per-organization isolation.

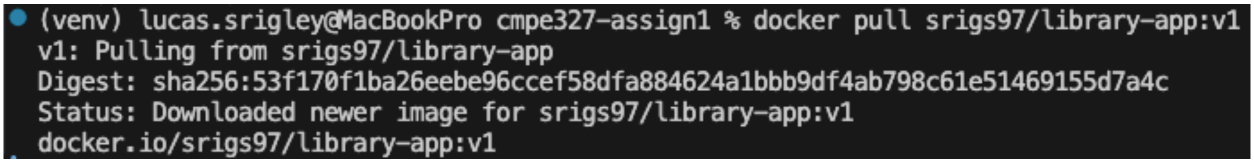
Get faster builds through shared caching across your team,

Delete



```
(venv) lucas.srigley@MacBookPro cmpe327-assign1 % docker rmi srigs97/library-app:v1
Untagged: srigs97/library-app:v1
```

Pull



```
(venv) lucas.srigley@MacBookPro cmpe327-assign1 % docker pull srigs97/library-app:v1
v1: Pulling from srigs97/library-app
Digest: sha256:53f170f1ba26eebe96ccef58dfa884624a1bbb9df4ab798c61e51469155d7a4c
Status: Downloaded newer image for srigs97/library-app:v1
docker.io/srigs97/library-app:v1
```

7. Challenges and Reflections

When writing the Playwright tests, I noticed that when rerunning tests, they were failing when they were originally passing previously. To fix this, I called the reset_db after each test suite run and it fixed my issue. Another issue I was having was the book ID was constantly being updated every time the website refreshed. I originally hardcoded the ID, but the tests kept failing. I solved this by dynamically fetching the current book ID value by finding the element on the page. Lastly, I was having issues with the CI and the test_e2e.py suite failing. This issue was resolved by starting the server in the CI file before running the tests.