327 A4

Sandy Mourad

20406862

Github repo: https://github.com/Sandy-Mourad/cisc327-a4-6862

E2E Testing approach:

I used playwright for python with pytest to run real browser-based end-to-end tests. The goal of this test was to verify that the most important user workflows in the library management system functions correctly when we execute it through the real browser interface rather than isolated backend calls. The reason I chose playwright was because it provides reliable and deterministic browser automation. It also integrates cleanly with pytest and is very good for web interfaces.

I started the test file with the flask application inside the test suite itself. A background thread then launches the app using the existing create_app() factory defined in my [app.py](app.py) with debug = false and use_reloader = False to prevent multiple processes. A session-scoped pytest fixture ensures the server is initialized only once for the entire test run. A little delay is added to guarantee that the server has fully started before Playwright attempts to load any pages. This approach eliminates all manual startup steps and provides a stable, repeatable test environment.

My user flow test was first. I opened the catalog page to confirm the application is running and then navigated to the add book page through the real ui and then filled out the add book form with a randomly generated valid ISBN and legitimate title or author values. Then submit the form and verify that a success message appears, then return to the catalog and confirm that the newly added book appears in the listing. Then, borrow the newly added book by entering a valid patron ID and verify that the success message is displayed. And finally, verify that the borrowing confirmation message appears onthe page.

As for ISBN uniqueness, this application enforces uniqueness for ISBN values and so to prevent the test from failing on any double or repeated executions then the test just generates a valid random 13-digit ISBN for each run which guarantees that the add book form passes validation and that the test remains stable even when executed multiple times.

This approach works because it tests a full vertical slice: UI, routing, business logic, templates, DB, and user-visible messages. Assertions also check both visible state and backend behaviour (talked about below). This mimics a real user, giving high confidence in the system's correctness.

The assertions i used:

1: catalog page loads: i checked that the catalog page displays expected text after navigation

2: successful add book form submission: after filling out the form and submitting it, I verified that a success message appeared in the page body

3: new book appears in the catalog: i returned to the catalog page and asserted that the newly added title and author were visible inthe table

4: correct row is used for borrowing: i selected the table row containing the newly added book and interaction with the borrow from inside that rowe

5: borrow confirmation message: after submitting the borrow action , i checked that "successfully borrowed” appeared on the page

Execution instruction:

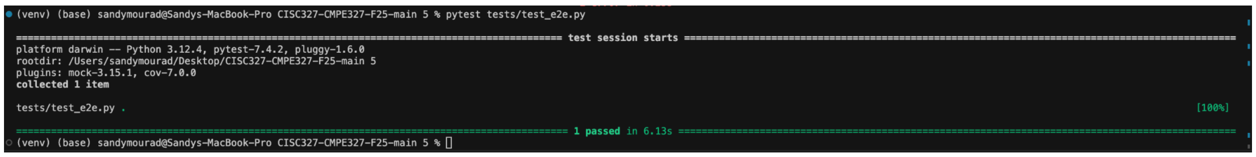
Create virtual environment: python -m venv venv

Open environment: source venv/bin/activate

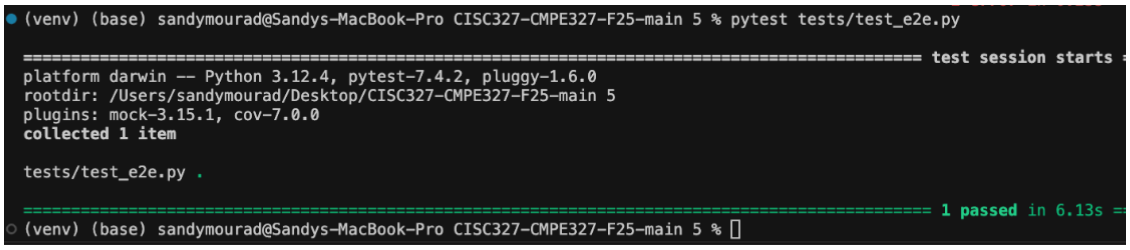
Install project Dependencies: pip install -r requirements.txt

Install Playwright browsers: python -m playwright install

Run e2e test file: pytest tests/test_e2e.py



```
(venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 % pytest tests/test_e2e.py

=================================================================================== test session starts ===================================================================================
platform darwin -- Python 3.12.4, pytest-7.4.2, pluggy-1.6.0
rootdir: /Users/sandymourad/Desktop/CISC327-CMPE327-F25-main 5
plugins: mock-3.15.1, cov-7.0.0
collected 1 item

tests/test_e2e.py .                                                                                                                                                                  [100%]

==================================================================================== 1 passed in 6.13s ====================================================================================
(venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 % 
```



```
(venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 % pytest tests/test_e2e.py

==================================================================================== test session starts
platform darwin -- Python 3.12.4, pytest-7.4.2, pluggy-1.6.0
rootdir: /Users/sandymourad/Desktop/CISC327-CMPE327-F25-main 5
plugins: mock-3.15.1, cov-7.0.0
collected 1 item

tests/test_e2e.py .

===================================================================================== 1 passed in 6.13s ==
(venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 % 
```

(Zoomed)

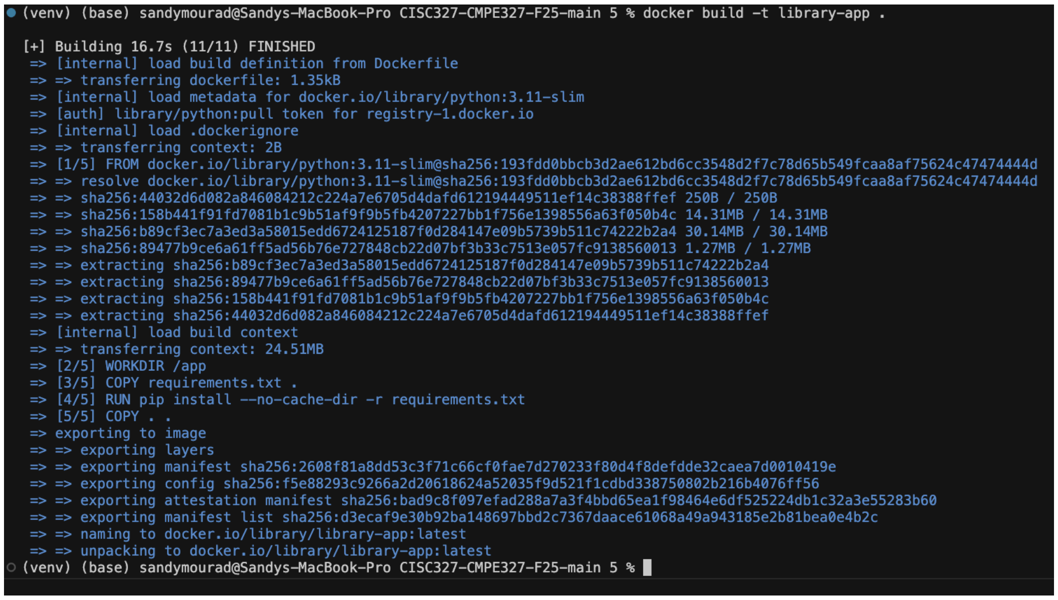
Test case summary:

Test case	**Actions performed**	**Expected results**
Test Case 1: Add Book - Catalog Verification	Navigate to /add_book Fill in title, author, random valid 13-digit ISBN, # of copies. Submit the add book form Navigate to /catalog and locate new book	A success flash message appears New book is visible in the catalog table The title, author, and ISBN match the values entered in the form The availability displays the correct initial value
Test Case 2: Borrow Book - Availability Update	Ensure the test book exists (added in previous step) Navigate to /catalog Enter valid patron ID Click the borrow button for that same book row	A successfully borrowed flash message appears The availability decreases by 1 The UI updates the correct table row (but only the selected books availability changes)

Dockerization Process:

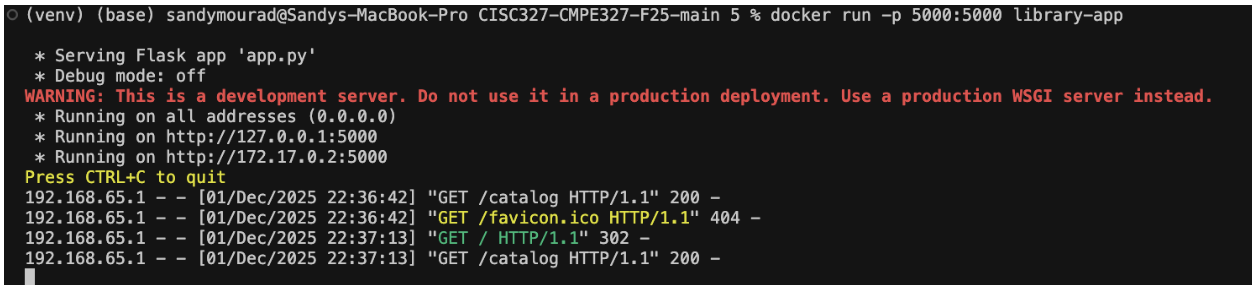
To containerize the flask application, i first created a docker file at the project root. The fockerfile installs all the dependencies and copies project files and exposes port 5000 and runs the application using python app.py

1: First I **created a dockerfile** with the required instructions : base image, install dependencies, copy files, expose port, run app.
2: Building Docker image: docker build -t library-app



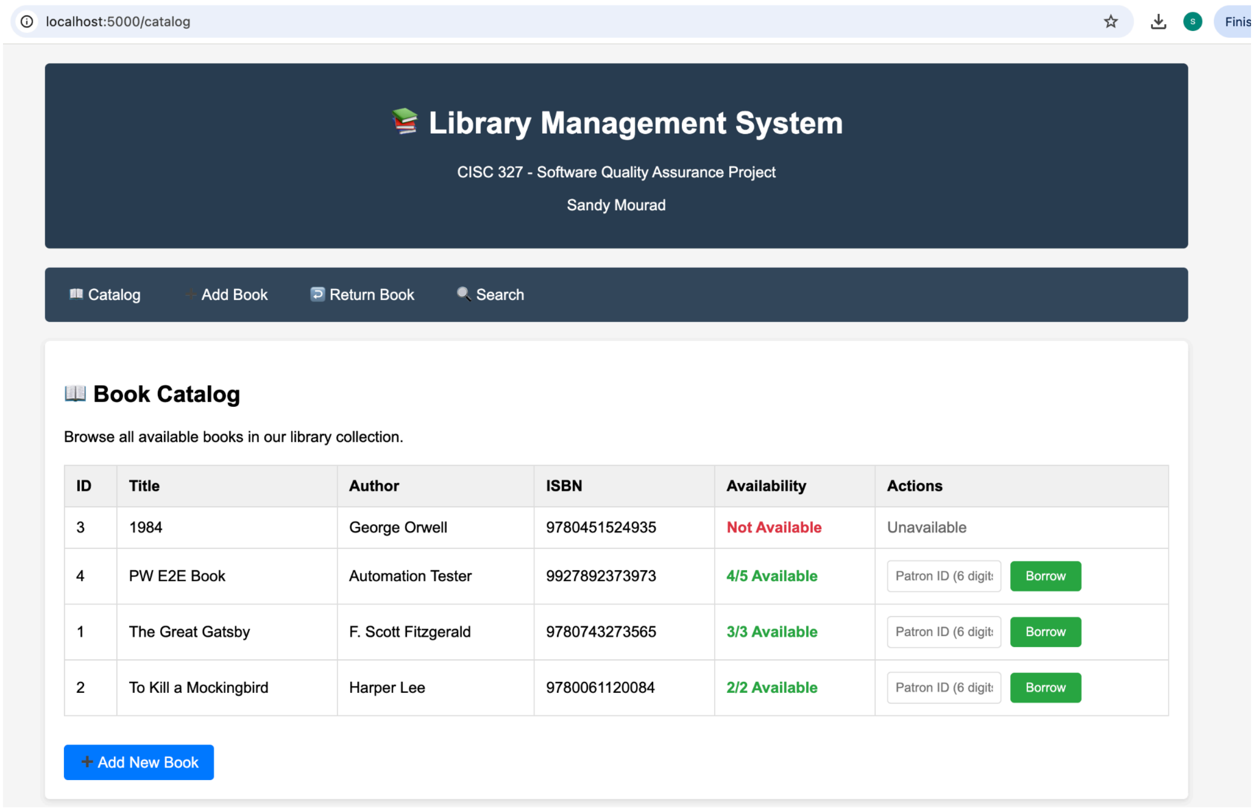
```
● (venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 % docker build -t library-app .

 [+] Building 16.7s (11/11) FINISHED
  => [internal] load build definition from Dockerfile
  => => transferring dockerfile: 1.35kB
  => [internal] load metadata for docker.io/library/python:3.11-slim
  => [auth] library/python:pull token for registry-1.docker.io
  => [internal] load .dockerignore
  => => transferring context: 2B
  => [1/5] FROM docker.io/library/python:3.11-slim@sha256:193fdd0bbcb3d2ae612bd6cc3548d2f7c78d65b549fcaa8af75624c47474444d
  => => resolve docker.io/library/python:3.11-slim@sha256:193fdd0bbcb3d2ae612bd6cc3548d2f7c78d65b549fcaa8af75624c47474444d
  => => sha256:44032d6d082a846084212c224a7e6705d4dafd612194449511ef14c38388ffef 250B / 250B
  => => sha256:158b441f91fd7081b1c9b51af9f9b5fb4207227bb1f756e1398556a63f050b4c 14.31MB / 14.31MB
  => => sha256:b89cf3ec7a3ed3a58015edd6724125187f0d284147e09b5739b511c74222b2a4 30.14MB / 30.14MB
  => => sha256:89477b9ce6a61ff5ad56b76e727848cb22d07bf3b33c7513e057fc9138560013 1.27MB / 1.27MB
  => => extracting sha256:b89cf3ec7a3ed3a58015edd6724125187f0d284147e09b5739b511c74222b2a4
  => => extracting sha256:89477b9ce6a61ff5ad56b76e727848cb22d07bf3b33c7513e057fc9138560013
  => => extracting sha256:158b441f91fd7081b1c9b51af9f9b5fb4207227bb1f756e1398556a63f050b4c
  => => extracting sha256:44032d6d082a846084212c224a7e6705d4dafd612194449511ef14c38388ffef
  => [internal] load build context
  => => transferring context: 24.51MB
  => [2/5] WORKDIR /app
  => [3/5] COPY requirements.txt .
  => [4/5] RUN pip install --no-cache-dir -r requirements.txt
  => [5/5] COPY . .
  => exporting to image
  => => exporting layers
  => => exporting manifest sha256:2608f81a8dd53c3f71c66cf0fae7d270233f80d4f8defdde32caea7d0010419e
  => => exporting config sha256:f5e88293c9266a2d20618624a52035f9d521f1cdbd338750802b216b4076ff56
  => => exporting attestation manifest sha256:bad9c8f097efad288a7a3f4bbd65ea1f98464e6df525224db1c32a3e55283b60
  => => exporting manifest list sha256:d3ecaf9e30b92ba148697bbd2c7367daace61068a49a943185e2b81bea0e4b2c
  => => naming to docker.io/library/library-app:latest
  => => unpacking to docker.io/library/library-app:latest
○ (venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 %
```

3: Running application in Docker: docker run -p 5000:5000 library-app



```
○ (venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 % docker run -p 5000:5000 library-app

  * Serving Flask app 'app.py'
  * Debug mode: off
 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
 192.168.65.1 - - [01/Dec/2025 22:36:42] "GET /catalog HTTP/1.1" 200 -
 192.168.65.1 - - [01/Dec/2025 22:36:42] "GET /favicon.ico HTTP/1.1" 404 -
 192.168.65.1 - - [01/Dec/2025 22:37:13] "GET / HTTP/1.1" 302 -
 192.168.65.1 - - [01/Dec/2025 22:37:13] "GET /catalog HTTP/1.1" 200 -
```

4: Verify that the app is running on local host: i visited http://localhost:5000/catalog



📚 Library Management System

CISC 327 - Software Quality Assurance Project

Sandy Mourad

📖 Book Catalog

Browse all available books in our library collection.

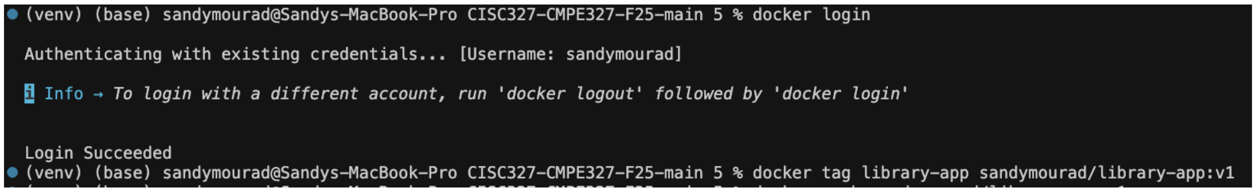
ID	Title	Author	ISBN	Availability	Actions
3	1984	George Orwell	9780451524935	Not Available	Unavailable
4	PW E2E Book	Automation Tester	9927892373973	4/5 Available	Patron ID (6 digit Borrow
1	The Great Gatsby	F. Scott Fitzgerald	9780743273565	3/3 Available	Patron ID (6 digit Borrow
2	To Kill a Mockingbird	Harper Lee	9780061120084	2/2 Available	Patron ID (6 digit Borrow

Docker hub deployment:

1: Then login to docker through terminal - docker login

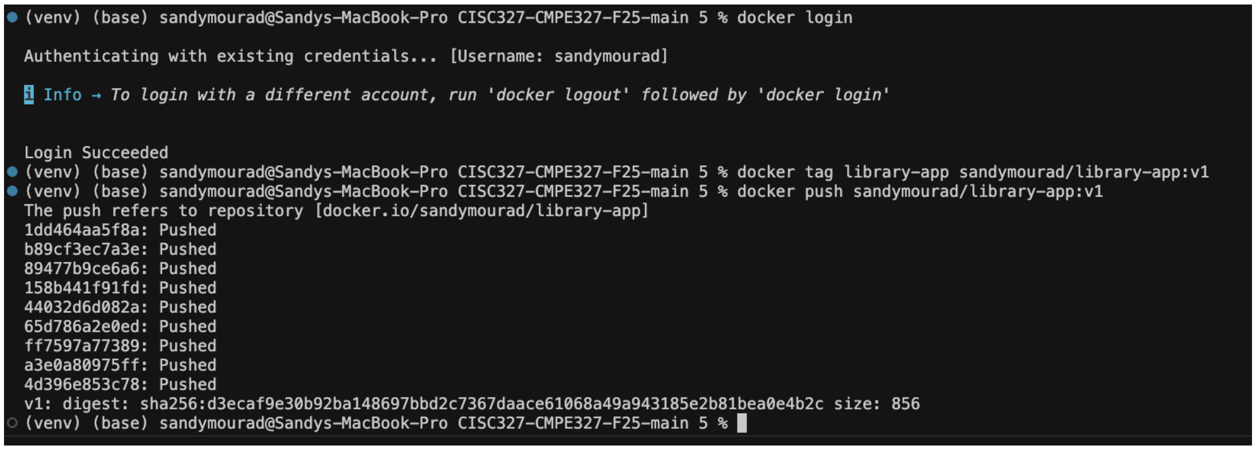
2: Then tag your image for docker hub: docker tag library-app sandymourad/library-app:v1

(2 in 1 image)



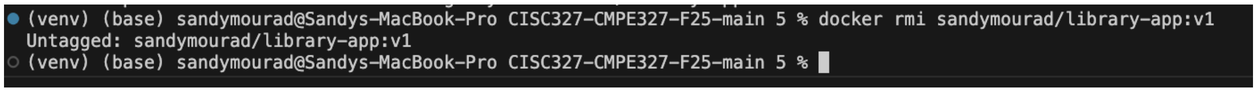
```
● (venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 % docker login

  Authenticating with existing credentials... [Username: sandymourad]

  i Info → To login with a different account, run 'docker logout' followed by 'docker login'


  Login Succeeded
● (venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 % docker tag library-app sandymourad/library-app:v1
```

3: Then push image for docker hub: docker push sandymourad/library-app:v1



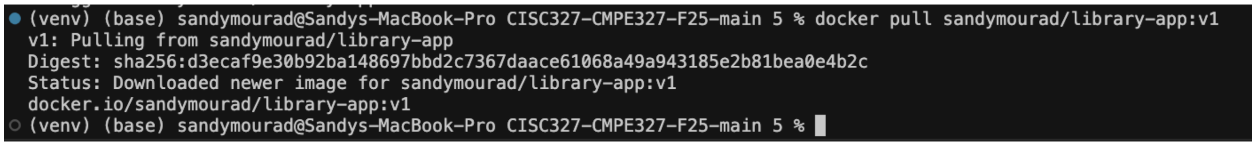
```
(venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 % docker login

Authenticating with existing credentials... [Username: sandymourad]

i Info → To login with a different account, run 'docker logout' followed by 'docker login'


Login Succeeded
(venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 % docker tag library-app sandymourad/library-app:v1
(venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 % docker push sandymourad/library-app:v1
The push refers to repository [docker.io/sandymourad/library-app]
1dd464aa5f8a: Pushed
b89cf3ec7a3e: Pushed
89477b9ce6a6: Pushed
158b441f91fd: Pushed
44032d6d082a: Pushed
65d786a2e0ed: Pushed
ff7597a77389: Pushed
a3e0a80975ff: Pushed
4d396e853c78: Pushed
v1: digest: sha256:d3ecaf9e30b92ba148697bbd2c7367daace61068a49a943185e2b81bea0e4b2c size: 856
(venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 %
```

4: Remove Local Image: docker rmi sandymourad/library-app:v1



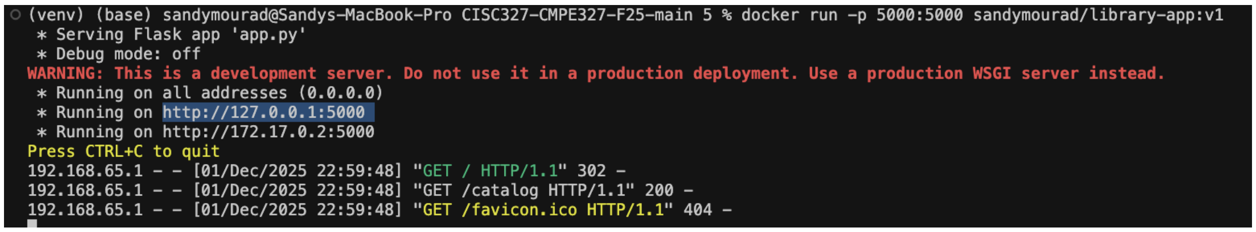
```
(venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 % docker rmi sandymourad/library-app:v1
Untagged: sandymourad/library-app:v1
(venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 %
```

5: Pull Image from Docker Hub: docker pull sandymourad/library-app:v1



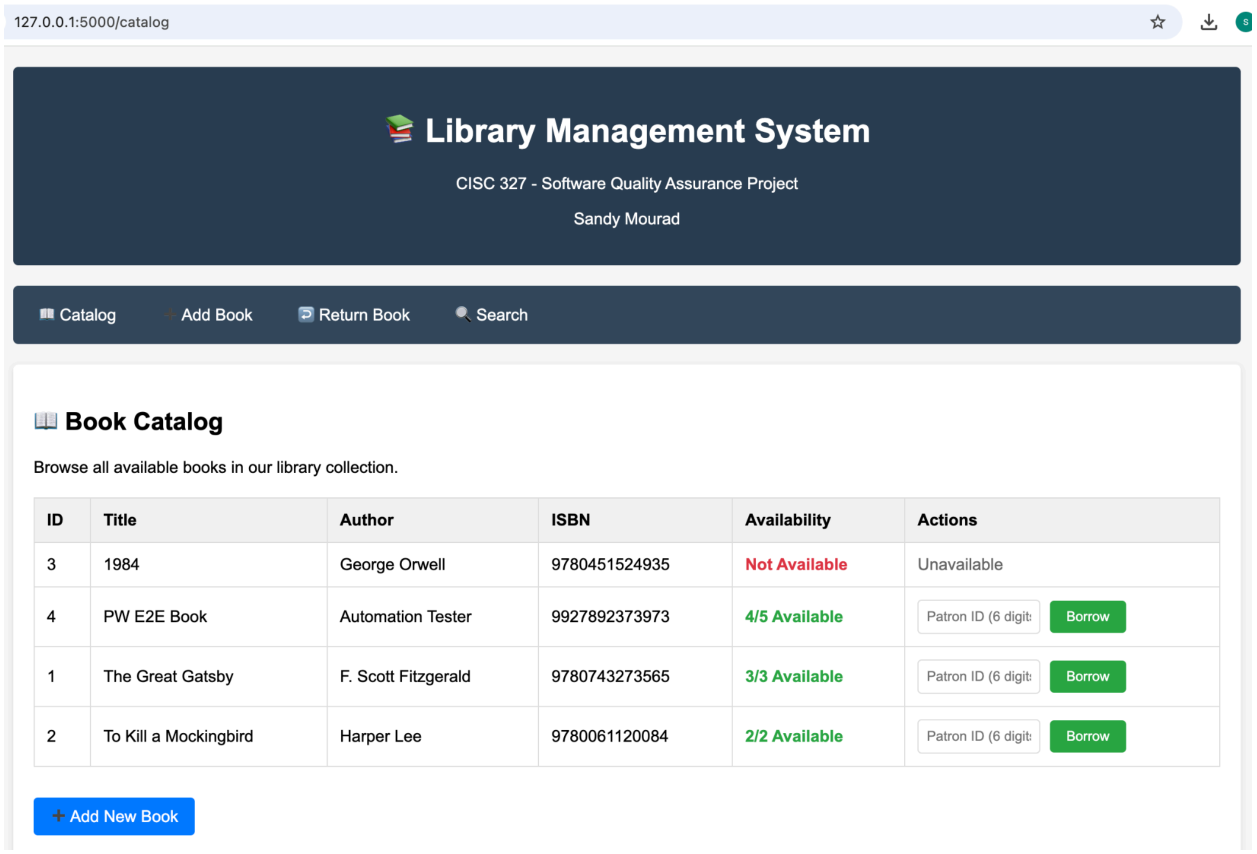
```
(venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 % docker pull sandymourad/library-app:v1
v1: Pulling from sandymourad/library-app
Digest: sha256:d3ecaf9e30b92ba148697bbd2c7367daace61068a49a943185e2b81bea0e4b2c
Status: Downloaded newer image for sandymourad/library-app:v1
docker.io/sandymourad/library-app:v1
(venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 %
```

6: Run the Docker Container: docker run -p 5000:5000 sandymourad/library-app:v1



```
(venv) (base) sandymourad@Sandys-MacBook-Pro CISC327-CMPE327-F25-main 5 % docker run -p 5000:5000 sandymourad/library-app:v1
 * Serving Flask app 'app.py'
 * Debug mode: off
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
192.168.65.1 - - [01/Dec/2025 22:59:48] "GET / HTTP/1.1" 302 -
192.168.65.1 - - [01/Dec/2025 22:59:48] "GET /catalog HTTP/1.1" 200 -
192.168.65.1 - - [01/Dec/2025 22:59:48] "GET /favicon.ico HTTP/1.1" 404 -
```

App works the same as we can see here:


127.0.0.1:5000/catalog

📚 Library Management System

CISC 327 - Software Quality Assurance Project

Sandy Mourad

📖 Catalog | Add Book | 🔄 Return Book | 🔍 Search

📖 Book Catalog

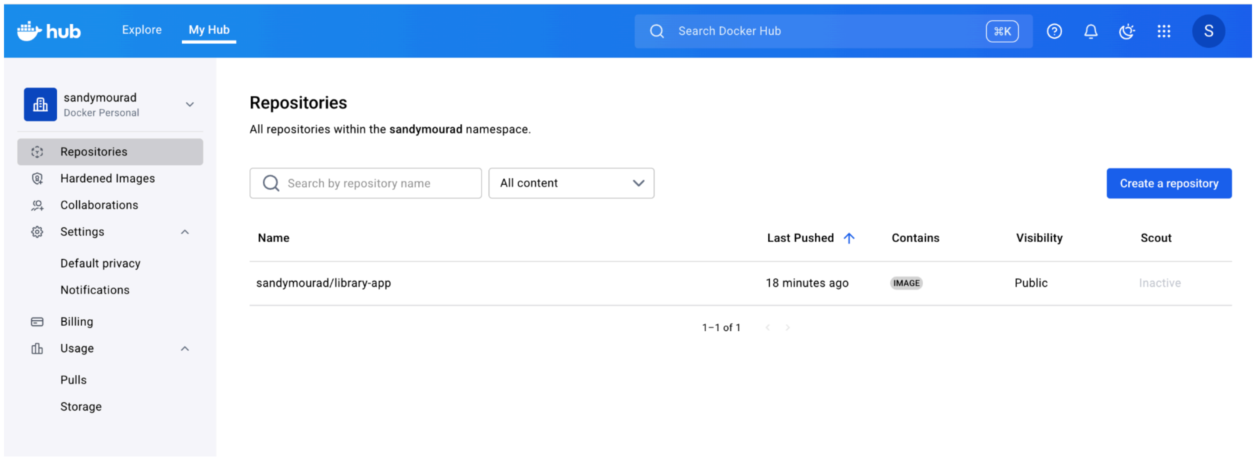
Browse all available books in our library collection.

ID	Title	Author	ISBN	Availability	Actions
3	1984	George Orwell	9780451524935	Not Available	Unavailable
4	PW E2E Book	Automation Tester	9927892373973	4/5 Available	Patron ID (6 digit: Borrow
1	The Great Gatsby	F. Scott Fitzgerald	9780743273565	3/3 Available	Patron ID (6 digit: Borrow
2	To Kill a Mockingbird	Harper Lee	9780061120084	2/2 Available	Patron ID (6 digit: Borrow

+ Add New Book

Confirmation below!

And here is my docker hub repository confirmation:

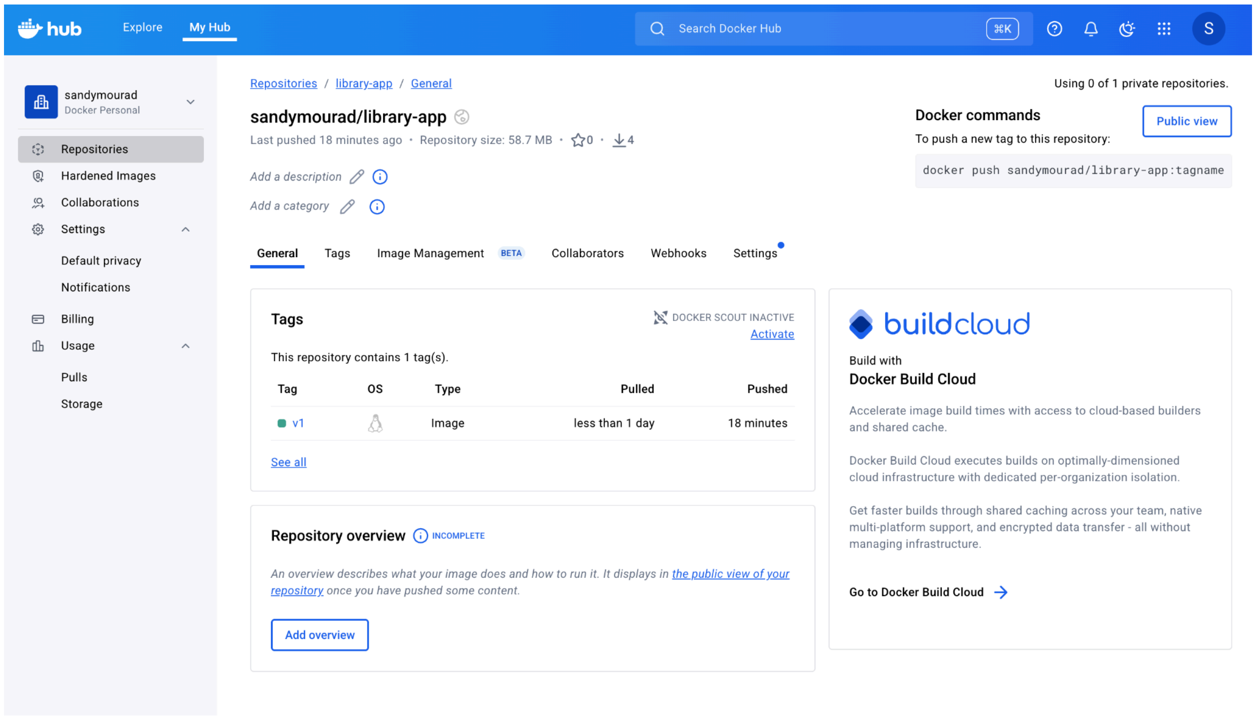


Repositories

All repositories within the **sandymourad** namespace.

Name	Last Pushed	Contains	Visibility	Scout
sandymourad/library-app	18 minutes ago	IMAGE	Public	Inactive

1–1 of 1



Repositories / library-app / General

Using 0 of 1 private repositories.

sandymourad/library-app

Last pushed 18 minutes ago • Repository size: 58.7 MB • ☆0 • ⤓4

Add a description

Add a category

Docker commands

To push a new tag to this repository:

```
docker push sandymourad/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	18 minutes

See all

Repository overview INCOMPLETE

An overview describes what your image does and how to run it. It displays in the public view of your repository once you have pushed some content.

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, native multi-platform support, and encrypted data transfer - all without managing infrastructure.