Assignment 4 Report

GIT: https://github.com/ethanwithap/cisc327-library-management-a4-7003.git

1. Student Information

Name: **Ethan Panikar**
Student ID: **20287003**
Date: **2025-11-30**

2. E2E Testing approach:

Tool: Playwright
Tested features:

Add Book and verify the catalogue

- Opens page, uses “Add Book”, and enters the correct ISBN and submits.
- After submission, it checks the catalogue to see if the new book is listed.

View the catalogue with sample data.

- Inserts three sample books using add_simple_data()
- Opens catalogue and asserts that these titles appear.

3. Execution Instructions

In the assignment folder:

1. **Load site:**

$env:FLASK_APP = "app:create_app"
flask run

2. **Install dependencies and run tests:**

pip install -r requirements.txt
pip install pytest playwright
python -m playwright install

pytest tests/test_e2e.py -v

Both tests will pass

3. **Build the Docker:**

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

4. **Docker Hub deployment:**

docker login

docker tag library-app ethanwithap/library-app:v1
docker push ethanwithap/library-app:v1

docker rmi ethanwithap/library-app:v1
docker pull ethanwithap/library-app:v1

docker run -p 5000:5000 ethanwithap/library-app:v1

4. Test Case Summary:

Test Case ID	User Flow	Steps	Expected result
TC-01	Add new book and verify catalog	Home > add book> fill form> submit> open catalog	New book title appears in catalog
TC-02	View catalog sample data	Load sample data > open catalog	Sample books appear

5. Dockerization Process:

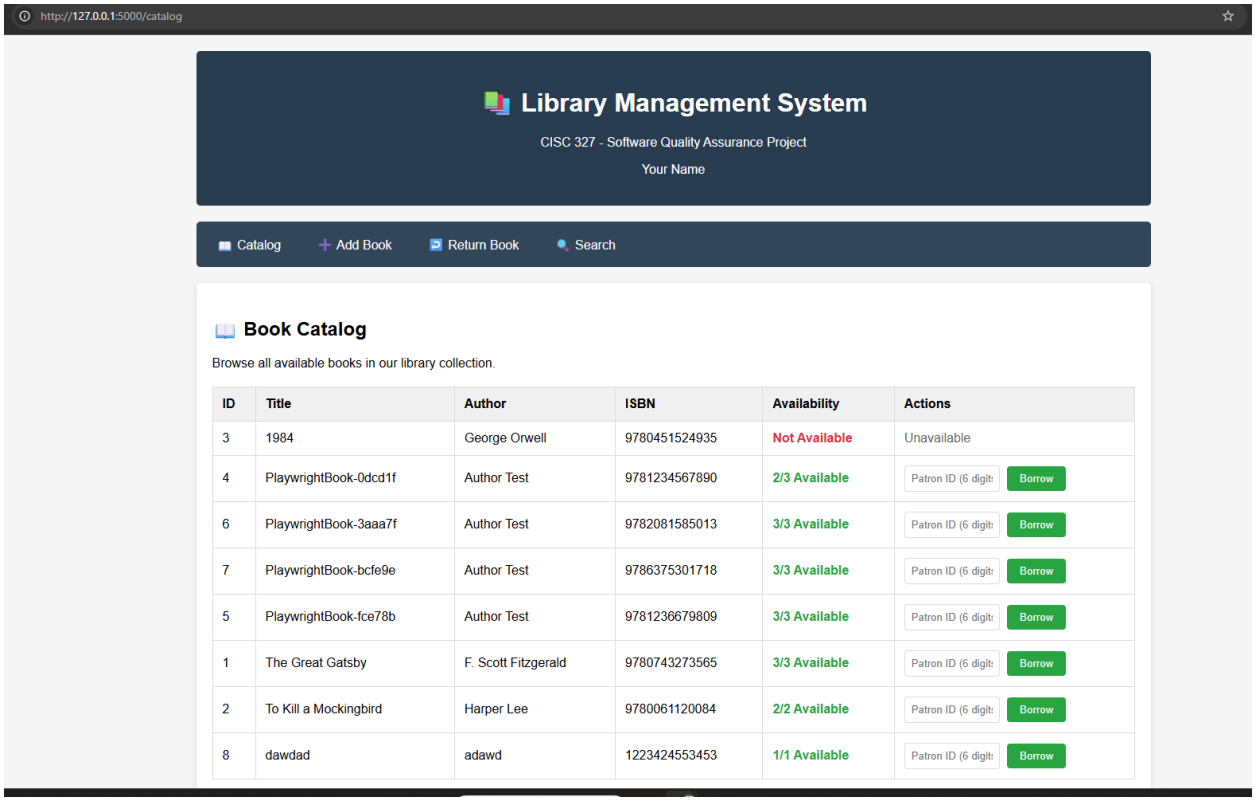
- Created the Dockerfile in the project root. It installs dependencies, copies the project files into the container, set the flash env and launches app using flask in port 5000.

```
FROM python:3.11-slim

ENV PYTHONDONTWRITEBYTECODE=1
ENV PYTHONUNBUFFERED=1

WORKDIR /app

COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

ENV FLASK_APP=app:create_app
ENV FLASK_RUN_HOST=0.0.0.0
ENV FLASK_RUN_PORT=5000

EXPOSE 5000

CMD ["flask", "run"]
```

  - From terminal, I built the image using the command: *docker build -t library-app .*

```
PS C:\Users\ethan\downloads\CISC327-CMPE327-F25-feature-assignment3\CISC327-CMPE327-F25-feature-assignment3> docker build -t library-app .
[+] Building 32.9s (11/11) FINISHED                                                                    docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                  0.0s
 => => transferring dockerfile: 350B                                                                                  0.0s
 => [internal] load metadata for docker.io/library/python:3.11-slim                                                   1.3s
 => [auth] library/python:pull token for registry-1.docker.io                                                         0.0s
 => [internal] load .dockerignore                                                                                     0.0s
 => => transferring context: 2B                                                                                       0.0s
 => [1/5] FROM docker.io/library/python:3.11-slim@sha256:193fdd0bbcb3d2ae612bd6cc3548d2f7c78d65b549fcaa8af75624c4     6.6s
 => => resolve docker.io/library/python:3.11-slim@sha256:193fdd0bbcb3d2ae612bd6cc3548d2f7c78d65b549fcaa8af75624c4     0.0s
 => => sha256:1771569cc1299abc9cc762fc4419523e721b11a3927ef968ae63ba0a4a88f2da 251B / 251B                            0.2s
 => => sha256:b3dd773c329649f22e467ae63d1c612a039a0559dec99ffb9ada904ab5c60c55 14.36MB / 14.36MB                      2.5s
 => => sha256:22b63e76fde1e200371ed9f3cee91161d192063bcff65c9ab6bf63819810a974 1.29MB / 1.29MB                        0.4s
 => => sha256:0e4bc2bd6656e6e004e3c749af70e5650bac2258243eb0949dea51cb8b7863db 29.78MB / 29.78MB                      4.0s
 => => extracting sha256:0e4bc2bd6656e6e004e3c749af70e5650bac2258243eb0949dea51cb8b7863db                             1.2s
 => => extracting sha256:22b63e76fde1e200371ed9f3cee91161d192063bcff65c9ab6bf63819810a974                             0.2s
 => => extracting sha256:b3dd773c329649f22e467ae63d1c612a039a0559dec99ffb9ada904ab5c60c55                             1.0s
 => => extracting sha256:1771569cc1299abc9cc762fc4419523e721b11a3927ef968ae63ba0a4a88f2da                             0.0s
 => [internal] load build context                                                                                     0.2s
 => => transferring context: 148.92kB                                                                                 0.1s
 => [2/5] WORKDIR /app                                                                                                0.2s
 => [3/5] COPY requirements.txt .                                                                                     0.1s
 => [4/5] RUN pip install --no-cache-dir -r requirements.txt                                                         13.9s
 => [5/5] COPY . .                                                                                                    0.1s
 => exporting to image                                                                                               10.3s
 => => exporting layers                                                                                               8.2s
 => => exporting manifest sha256:c4a744edf8ef01aa48fb0930fcf47b63da51409a36944d235876029d289546e8                     0.0s
 => => exporting config sha256:1e2c028d739fbf167e8fec769fee641c289ceaf52826ddfd14fd6ad4837e1d82                       0.0s
 => => exporting attestation manifest sha256:2884914b3d85c9dabaf98a5cff1182ce0d5a68c8b4b6bd5e908fd35ba0212d1a         0.0s
 => => exporting manifest list sha256:b92b058278ad96aa27e660ad16bb711e99b9179d1cf82ebee405ecdf0fdcfa98                0.0s
 => => naming to docker.io/library/library-app:latest                                                                 0.0s
 => => unpacking to docker.io/library/library-app:latest                                                              1.9s
PS C:\Users\ethan\downloads\CISC327-CMPE327-F25-feature-assignment3\CISC327-CMPE327-F25-feature-assignment3> docker run -p 5000:5000 library-app
 * Serving Flask app 'app:create_app'
```

Clicking on the hosted local site brings you to the library management system page



6. Docker Hub Deployment

Next, I logged in to Docker in PowerShell and then tagged the image with my Docker account and then pushed the container using

- *docker tag library-app ethanwithap/library-app:v1*
- *docker push ethanwithap/library-app:v1*

```
Login Succeeded
PS C:\Users\ethan\downloads\CISC327-CMPE327-F25-feature-assignment3\CISC327-CMPE327-F25-feature-assignment3> docker tag
library-app ethanwithap/library-app:v1
PS C:\Users\ethan\downloads\CISC327-CMPE327-F25-feature-assignment3\CISC327-CMPE327-F25-feature-assignment3> docker imag
es
                                                                                                    i Info →  U In Use
IMAGE                          ID             DISK USAGE   CONTENT SIZE   EXTRA
ethanwithap/library-app:v1     3c3a7547e509        391MB         96.9MB    U
library-app:latest             3c3a7547e509        391MB         96.9MB    U
PS C:\Users\ethan\downloads\CISC327-CMPE327-F25-feature-assignment3\CISC327-CMPE327-F25-feature-assignment3> docker push
 ethanwithap/library-app:v1
The push refers to repository [docker.io/ethanwithap/library-app]
13f79e183e73: Pushed
9b2b6490f9dd: Pushed
b3dd773c3296: Layer already exists
02883e9e289e: Layer already exists
0169f4ef95b7: Layer already exists
22b63e76fde1: Layer already exists
aebb7b91e920: Layer already exists
1771569cc129: Layer already exists
0e4bc2bd6656: Layer already exists
v1: digest: sha256:3c3a7547e509bf41302c7254ab52f8f11bdcea3835cfe106acc3852d676d017b size: 856
```

Deleted image locally:

- *docker rmi ethanwithap/library-app:v1*

```
PS C:\Users\ethan\downloads\CISC327-CMPE327-F25-feature-assignment3\CISC327-CMPE327-F25-feature-assignment3> docker rmi
ethanwithap/library-app:v1
```

Lastly, I pulled the image and ran it again to test it:

- *docker pull ethanwithap/library-app:v1*

```
PS C:\Users\ethan\downloads\CISC327-CMPE327-F25-feature-assignment3\CISC327-CMPE327-F25-feature-assignment3> docker pull
 ethanwithap/library-app:v1
v1: Pulling from ethanwithap/library-app
Digest: sha256:3c3a7547e509bf41302c7254ab52f8f11bdcea3835cfe106acc3852d676d017b
Status: Downloaded newer image for ethanwithap/library-app:v1
docker.io/ethanwithap/library-app:v1
PS C:\Users\ethan\downloads\CISC327-CMPE327-F25-feature-assignment3\CISC327-CMPE327-F25-feature-assignment3> docker run
-p 5000:5000 ethanwithap/library-app:v1
 * Serving Flask app 'app:create_app'
 * 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
```

7. Challenges and Reflections

One challenge I faced was using Playwright for my end-to-end test cases. Some input fields did not have stable names/IDs, which caused Playwright to fail when trying to locate items. I fixed this by remaking my E2E tests to focus only on guaranteed versions of the assignment.

Another challenge I had was setting up Docker and understanding how it works/ what it does, and how to set it up. It was quite a steep learning curve, but anything I was unsure about, there was more than enough documentation online to guide me through the process. I learned that not all Web UIs are built with testability in mind. E2E seems to require selecting only certain real user flows rather than obscure ones.
This assignment really helped me understand automated testing and taught me how to use two new resources, Docker and Playwright.