1. Chris Jakubowski
   20388138
   11/25/25
2. For E2E testing, I tested two different features. Adding a book (with randomly generated name/isbn) and then asserted/verified the successful add message appeared, and using the search tool to search for a book using the partial title and asserted/verified the correct book displayed.
3. To run the tests the code is:
   pytest tests/test_e2e.py
   For the docker container the code to build and then run is:
   docker build -t library-app .
   docker run -p 5000:5000 library-app
4.

Test Case	Actions	Expected Results
test_add_new_book	Go to site, then the “Add Book” page. Fill in the table with random title/isbn and click the “Add Book to Catalog” button. Verify if the success message appeared	Book gets added to catalog list and successful add message is at top of page
test_search_for_book	Go to site, then the “Search” page. Fill in the search query with “Great Gatsby” and select the search type for partial match title. Click the search button. Verify the book “The Great Gatsby” appears to the user	The book “The Great Gatsby” is displayed at the bottom of the page

5. I created a Dockerfile and filled it in with the necessary information. I added playwright to the requirements.txt file which is referenced in the Dockerfile. I then downloaded Docker and made an account
   Screenshot of it being built:

```
$ docker build -t library-app .
[+] Building 58.2s (11/11) FINISHED                                                                  docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                 0.1s
 => => transferring dockerfile: 212B                                                                                 0.0s
 => [internal] load metadata for docker.io/library/python:3.11-slim                                                  1.7s
 => [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@sha256:193fdd0bbcb3d2ae612bd6cc3548d2f7c78d65b549fcaa8af756       17.4s
 => => resolve docker.io/library/python:3.11-slim@sha256:193fdd0bbcb3d2ae612bd6cc3548d2f7c78d65b549fcaa8af7562       0.0s
 => => sha256:1771569cc1299abc9cc762fc4419523e721b11a3927ef968ae63ba0a4a88f2da 251B / 251B                           0.1s
 => => sha256:b3dd773c329649f22e467ae63d1c612a039a0559dec99ffb9ada904ab5c60c55 14.36MB / 14.36MB                     8.3s
 => => sha256:22b63e76fde1e200371ed9f3cee91161d192063bcff65c9ab6bf63819810a974 1.29MB / 1.29MB                       1.1s
 => => sha256:0e4bc2bd6656e6e004e3c749af70e5650bac2258243eb0949dea51cb8b7863db 29.78MB / 29.78MB                    13.6s
 => => extracting sha256:0e4bc2bd6656e6e004e3c749af70e5650bac2258243eb0949dea51cb8b7863db                            1.9s
 => => extracting sha256:22b63e76fde1e200371ed9f3cee91161d192063bcff65c9ab6bf63819810a974                            0.2s
 => => extracting sha256:b3dd773c329649f22e467ae63d1c612a039a0559dec99ffb9ada904ab5c60c55                            1.3s
 => => extracting sha256:1771569cc1299abc9cc762fc4419523e721b11a3927ef968ae63ba0a4a88f2da                            0.0s
 => [internal] load build context                                                                                   29.3s
 => => transferring context: 137.51MB                                                                               29.3s
 => [2/5] WORKDIR /app                                                                                               0.3s
 => [3/5] COPY requirements.txt .                                                                                    0.1s
 => [4/5] RUN pip install --no-cache-dir -r requirements.txt                                                        17.6s
 => [5/5] COPY . .                                                                                                   0.4s
 => exporting to image                                                                                               8.7s
 => => exporting layers                                                                                              6.1s
 => => exporting manifest sha256:0658168e8412e294d4fbf791bf85c1747f83ae1f586253e2d00e49b5615fc0d5                    0.0s
 => => exporting config sha256:483cd32738dd7d9bdf2bbf51f4a9601f488177667b6171c86bd1bbdadc46fabb                      0.0s
 => => exporting attestation manifest sha256:807124095819fa5dc1e28665e806a0f88dd98215d2903462d6dc4530bbd8b09a       0.0s
 => => exporting manifest list sha256:985e0a091d3cbcd372fb6291ba07f7c6164d19efe013e5a8f7ff8e3ddd29f3de               0.0s
 => => naming to docker.io/library/library-app:latest                                                                0.0s
 => => unpacking to docker.io/library/library-app:latest                                                             2.4s
```

Screenshot of it running:

```
$ docker run -p 5000:5000 library-app
 * 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: 136-562-983
172.17.0.1 - - [25/Nov/2025 05:44:37] "GET / HTTP/1.1" 302 -
172.17.0.1 - - [25/Nov/2025 05:44:37] "GET /catalog HTTP/1.1" 200 -
172.17.0.1 - - [25/Nov/2025 05:44:37] "GET /favicon.ico HTTP/1.1" 404 -
```

6. It being pushed:

```
$ docker push chrisj98/library-app:v1
The push refers to repository [docker.io/chrisj98/library-app]
435a4557f599: Pushed
95ad0b29bfde: Pushed
21d5a2452fc2: Pushed
22b63e76fde1: Pushed
1771569cc129: Pushed
1cf8fbc4f04c: Pushed
b3dd773c3296: Pushed
0e4bc2bd6656: Pushed
1ef971e62984: Pushed
v1: digest: sha256:985e0a091d3cbcd372fb6291ba07f7c6164d19efe013e5a8f7ff8e3ddd29f3de size: 856
```

It being deleted:

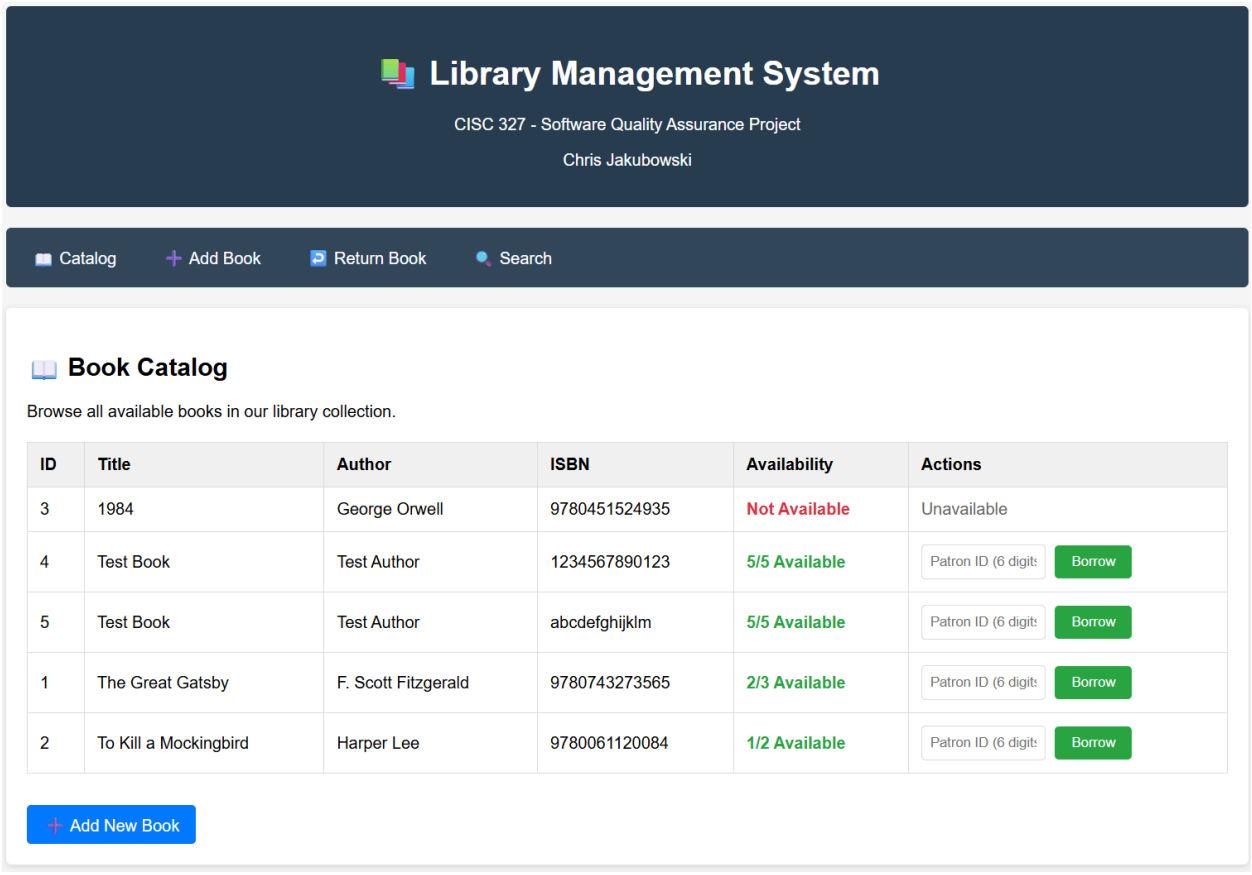
```
$ docker rmi chrisj98/library-app:v1
Untagged: chrisj98/library-app:v1
```

Being pulled:

```
$ docker pull chrisj98/library-app:v1
v1: Pulling from chrisj98/library-app
Digest: sha256:985e0a091d3cbcd372fb6291ba07f7c6164d19efe013e5a8f7ff8e3ddd29f3de
Status: Downloaded newer image for chrisj98/library-app:v1
docker.io/chrisj98/library-app:v1
```

Running:

```
$ docker run -p 5000:5000 chrisj98/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 instea
.
 * 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: 102-869-017
172.17.0.1 - - [25/Nov/2025 06:13:22] "GET / HTTP/1.1" 302 -
172.17.0.1 - - [25/Nov/2025 06:13:22] "GET /catalog HTTP/1.1" 200 -
```



7. The main challenge was learning about E2E testing. After getting a grasp on it, it was not too difficult to create the test functions. Figuring out the contents of a Dockerfile was also confusing, but brief research gave me the information I needed. Using docker itself didn't pose any challenge. I am unsure why when running the app through docker displays a time in the terminal 5 hours in the future.