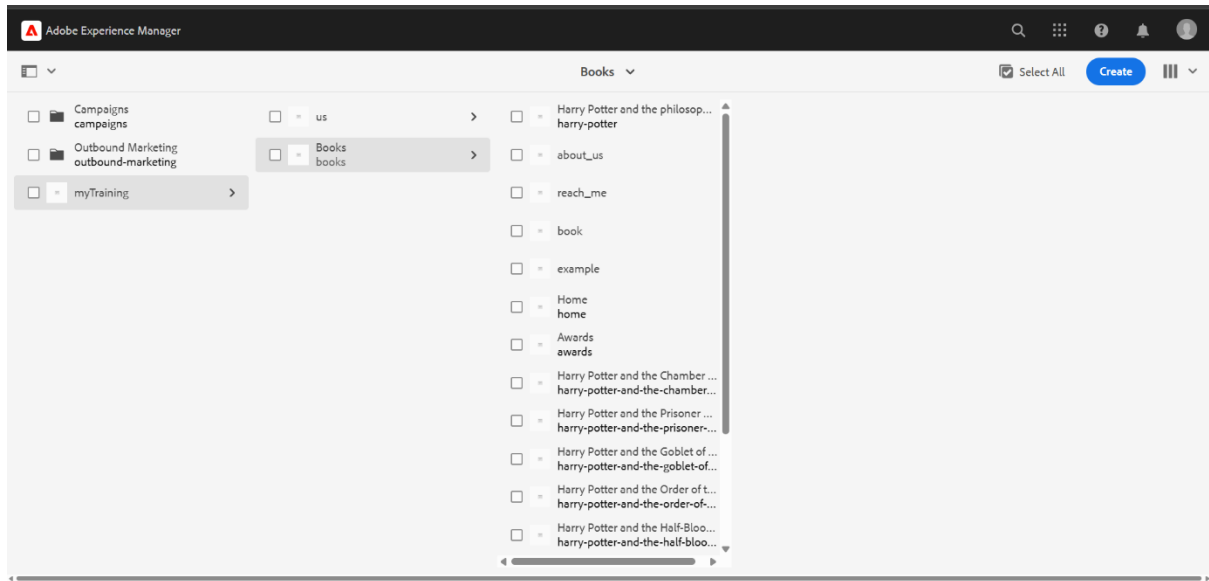
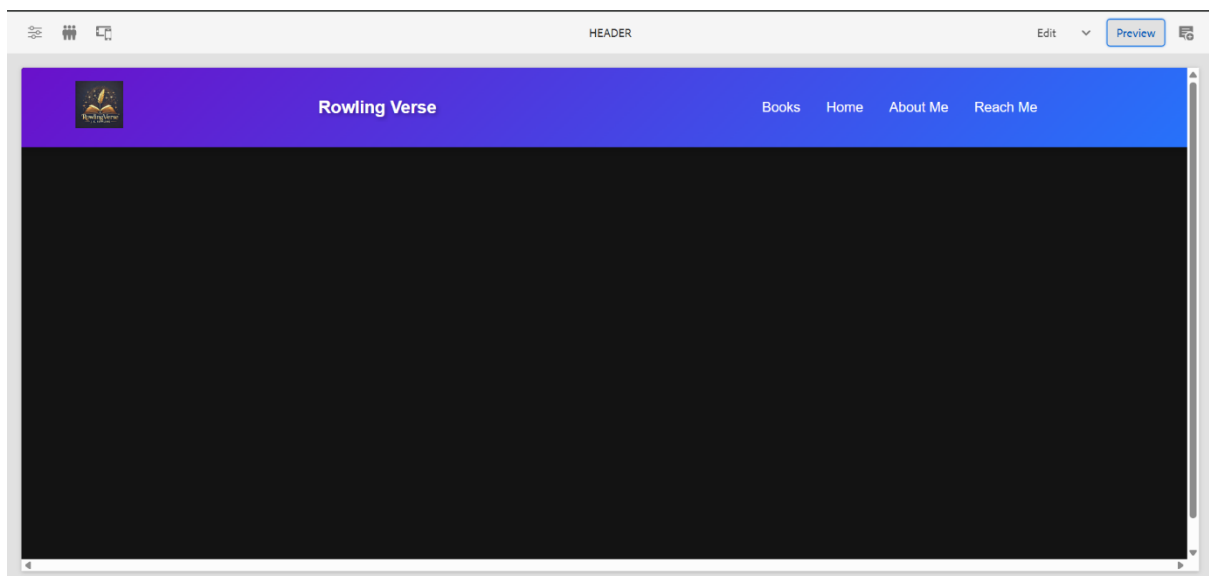


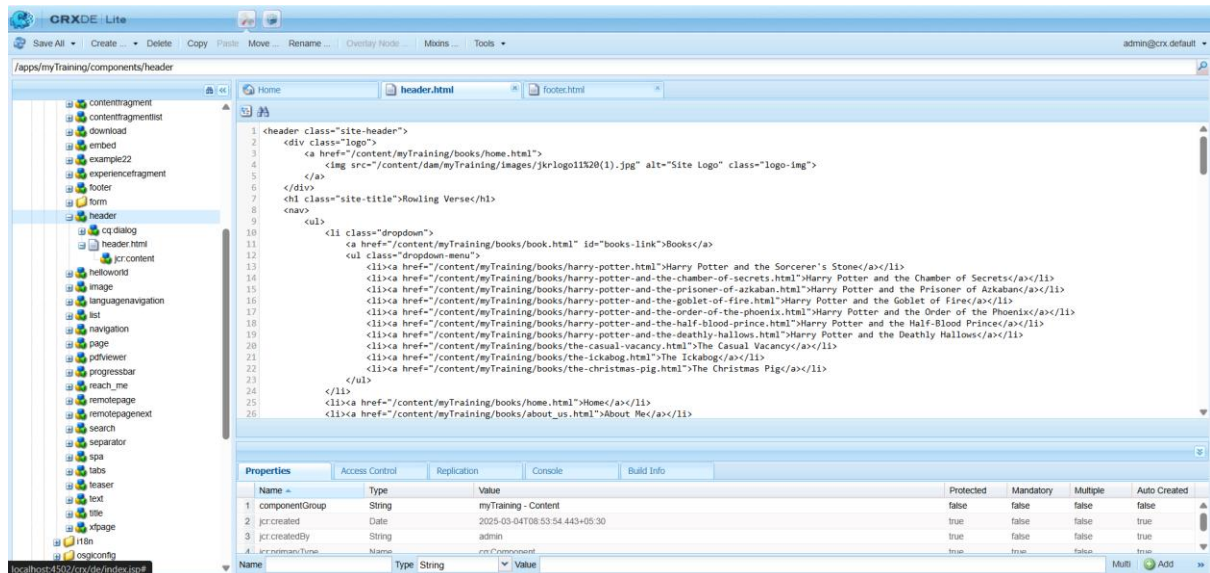
Pages created:(myTraining/books/):



Header preview:

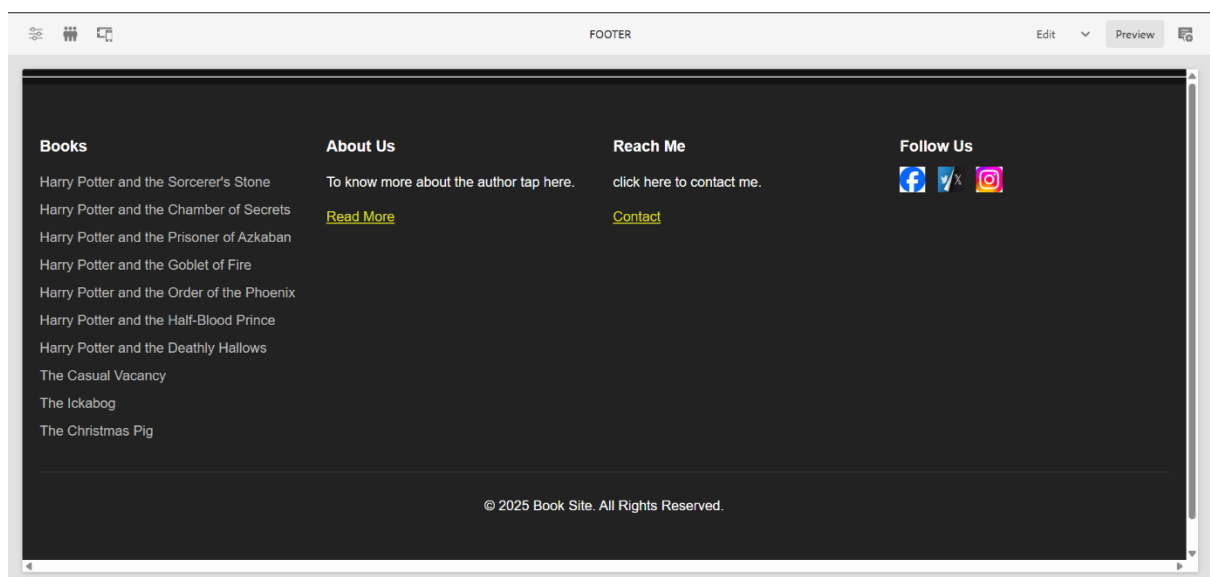


Code for header:

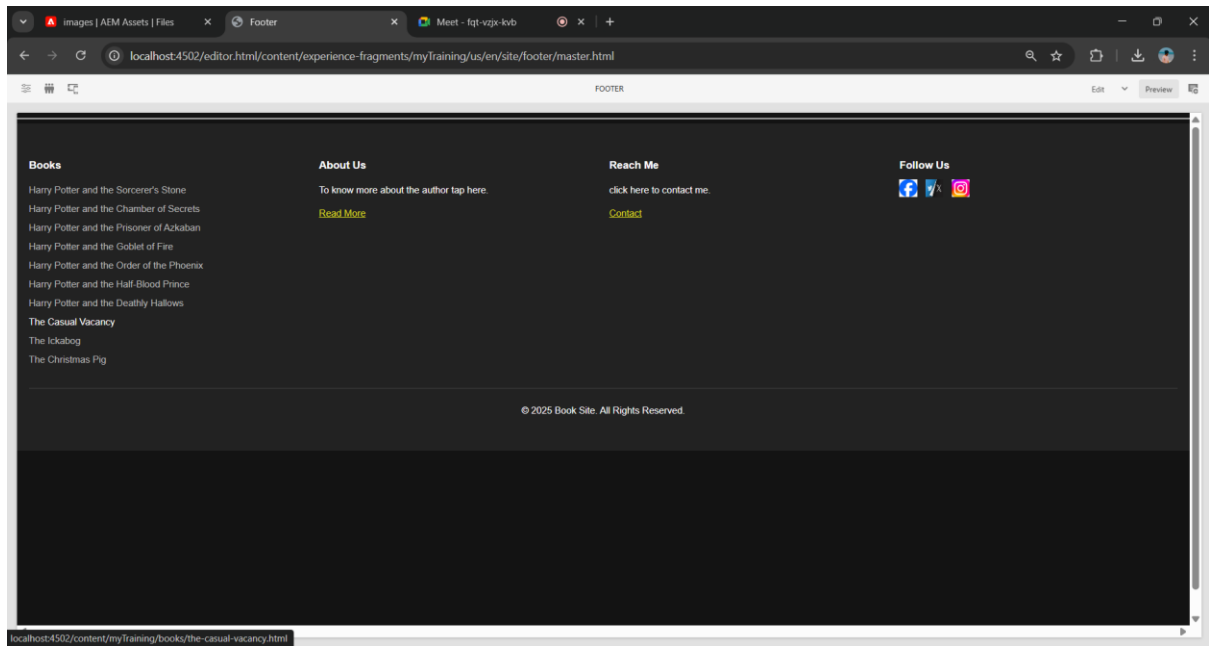


Footer preview:

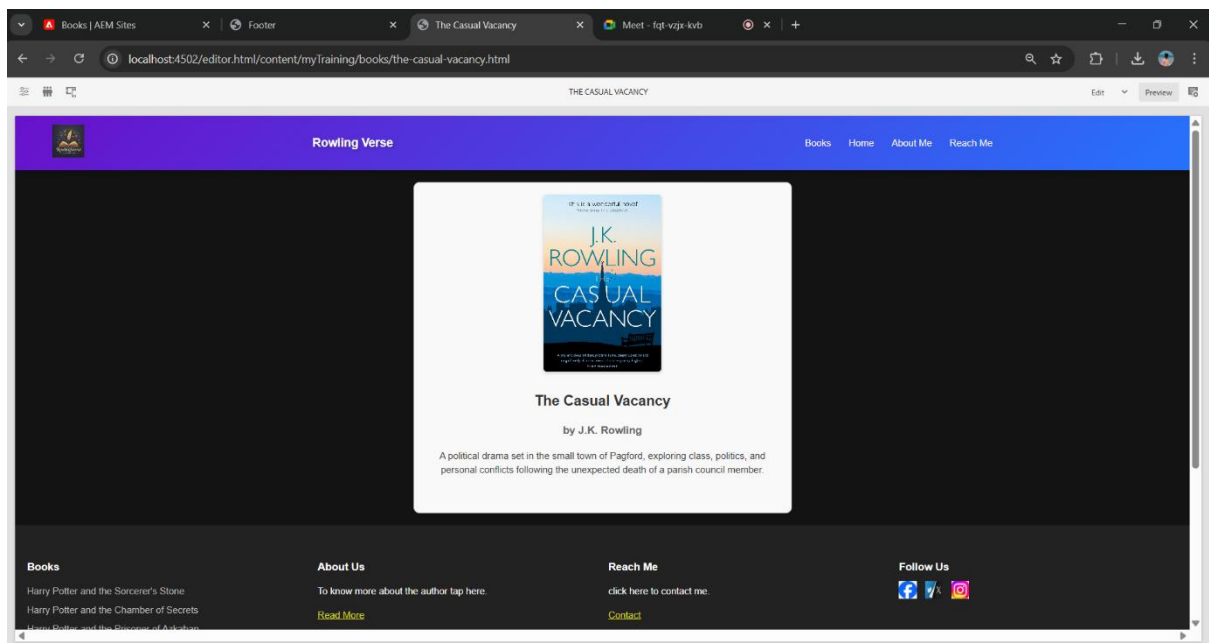
Footer contains books list(move to specific book page on tapping any books),about me,reach me and social media links.



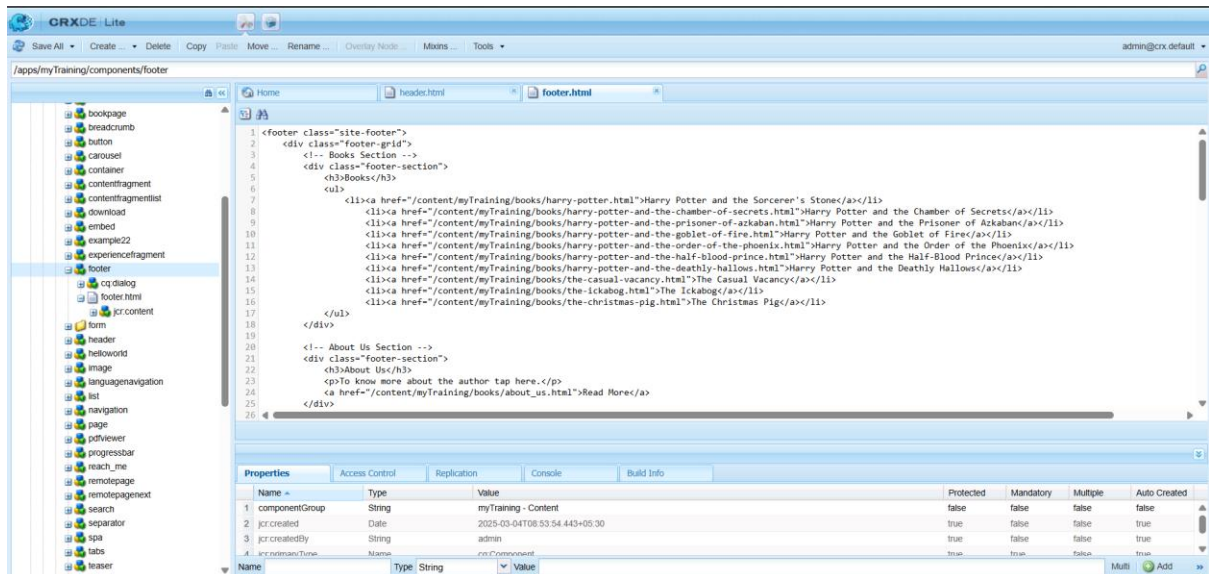
On tapping books on footer, it'll open that book in new tab:



This book is opened in a new tab:

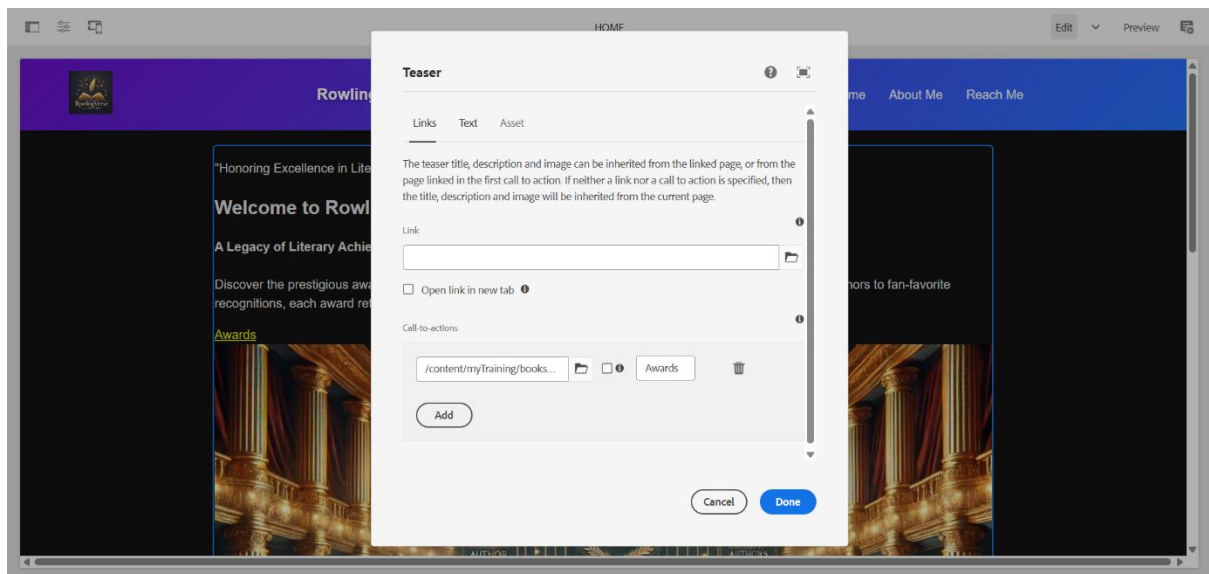


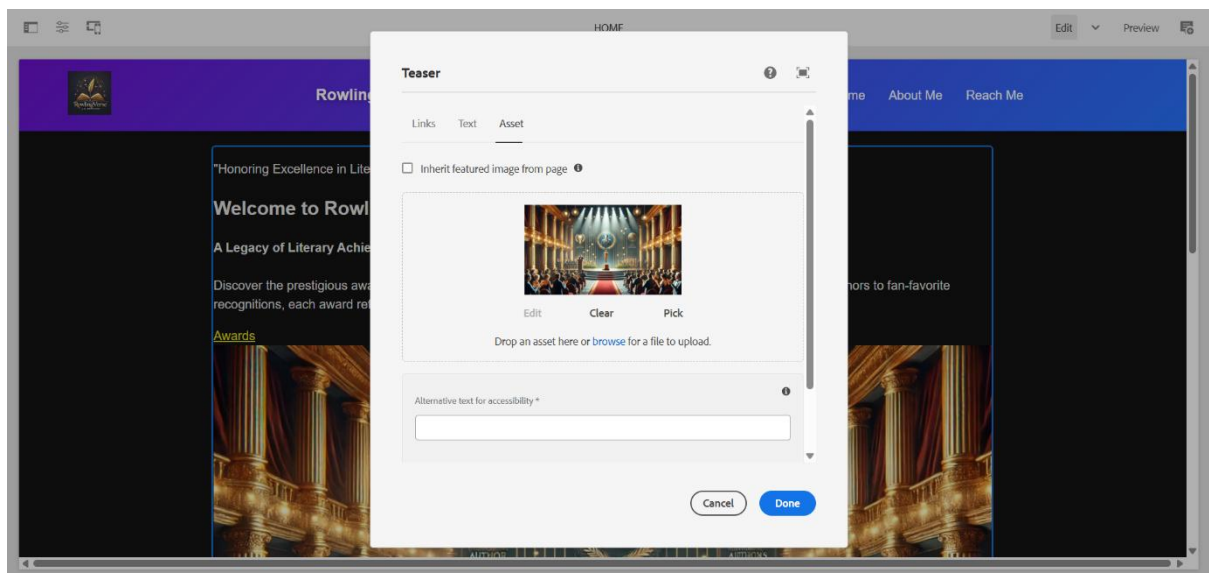
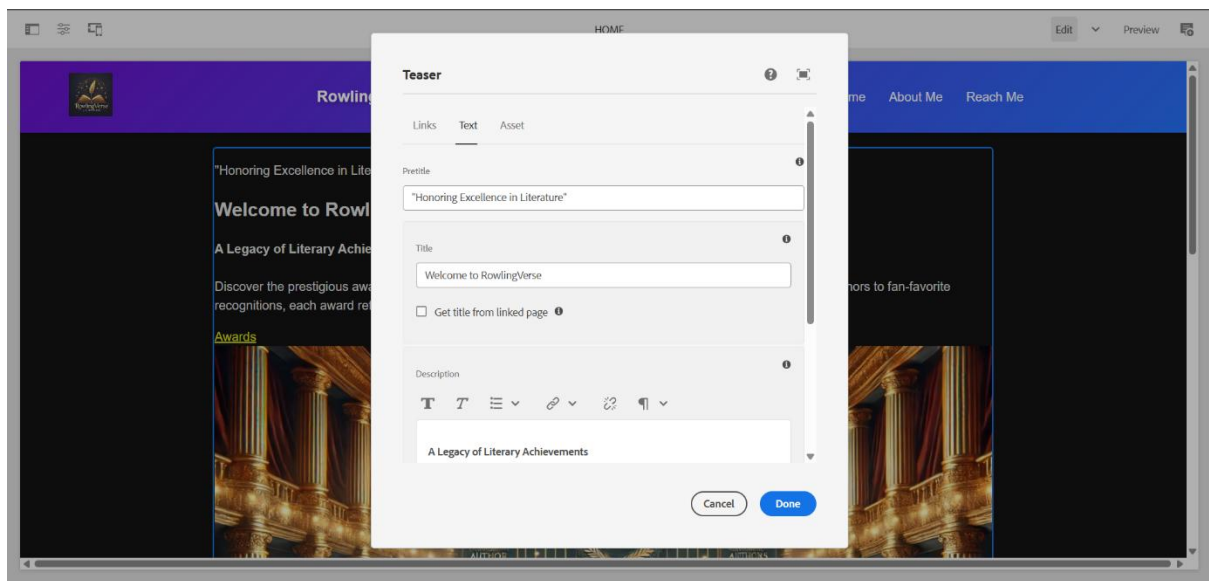
Code for Footer:

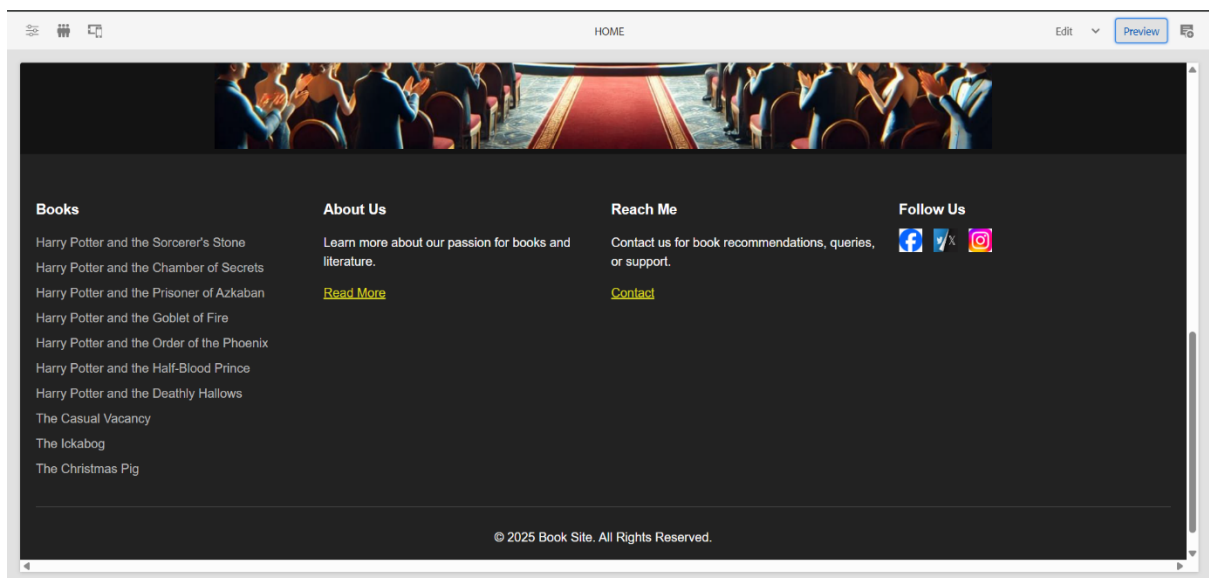
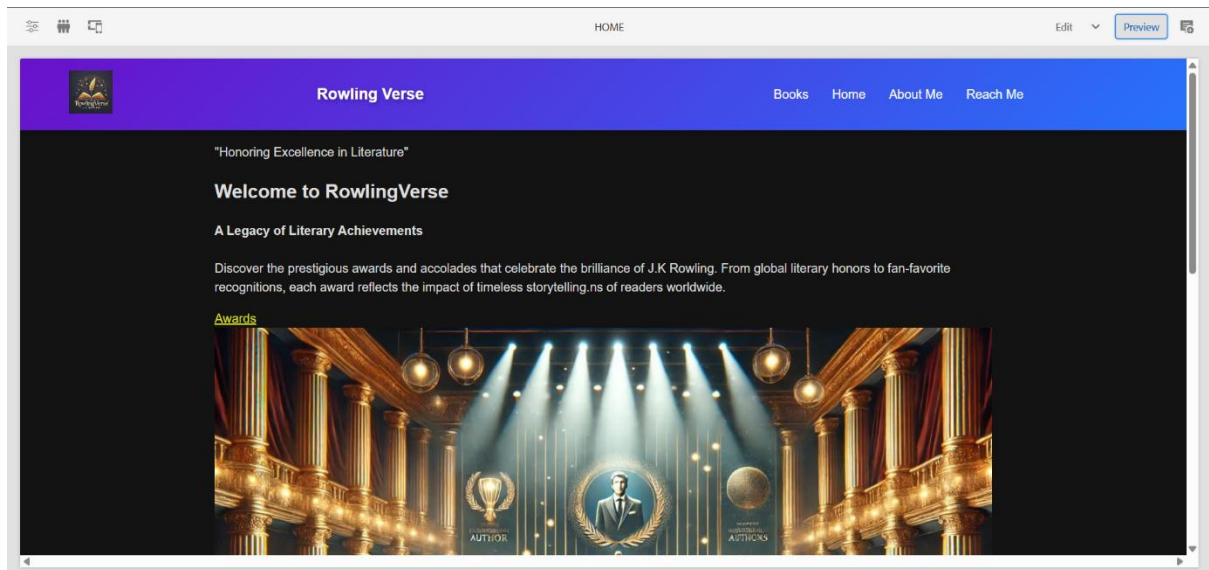


Home page:(created using teaser component):

Created one CTA that opens awards page and provided texts and assets.

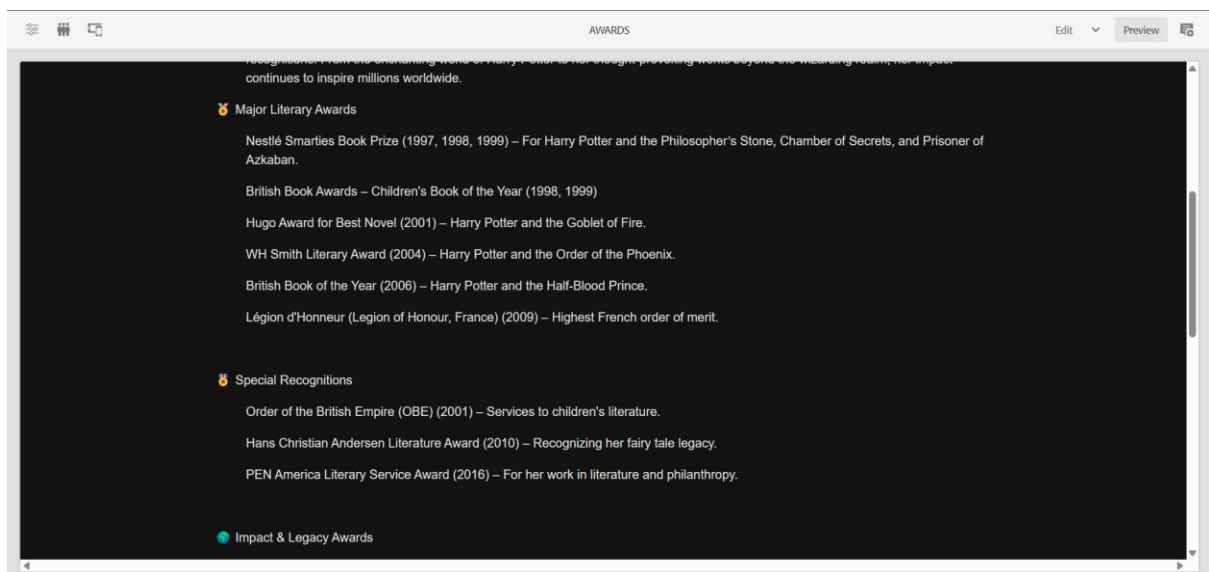
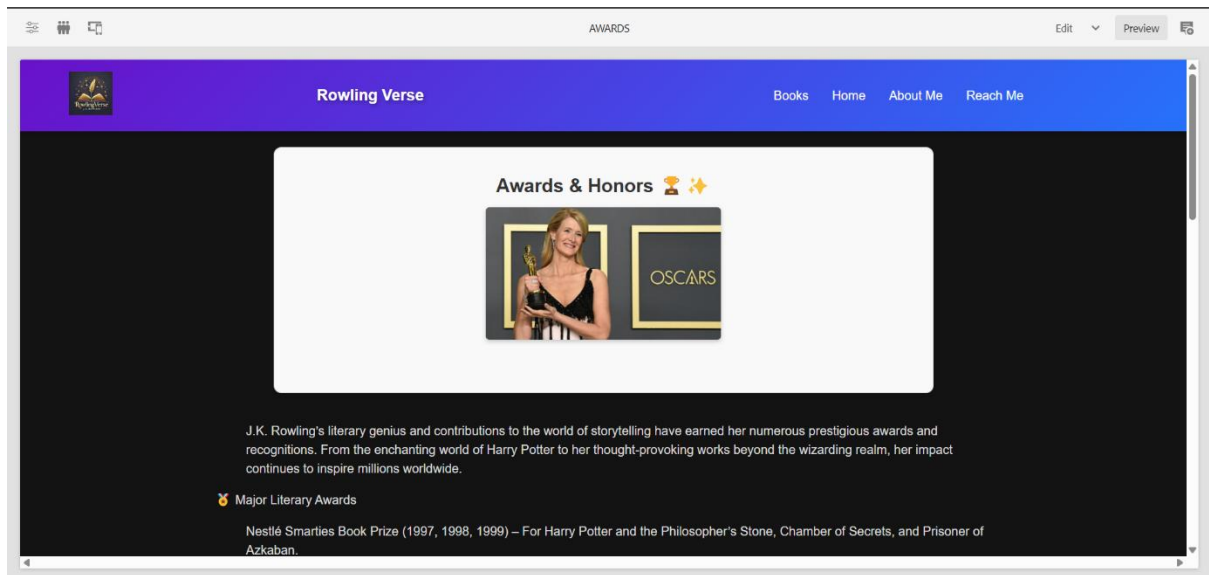


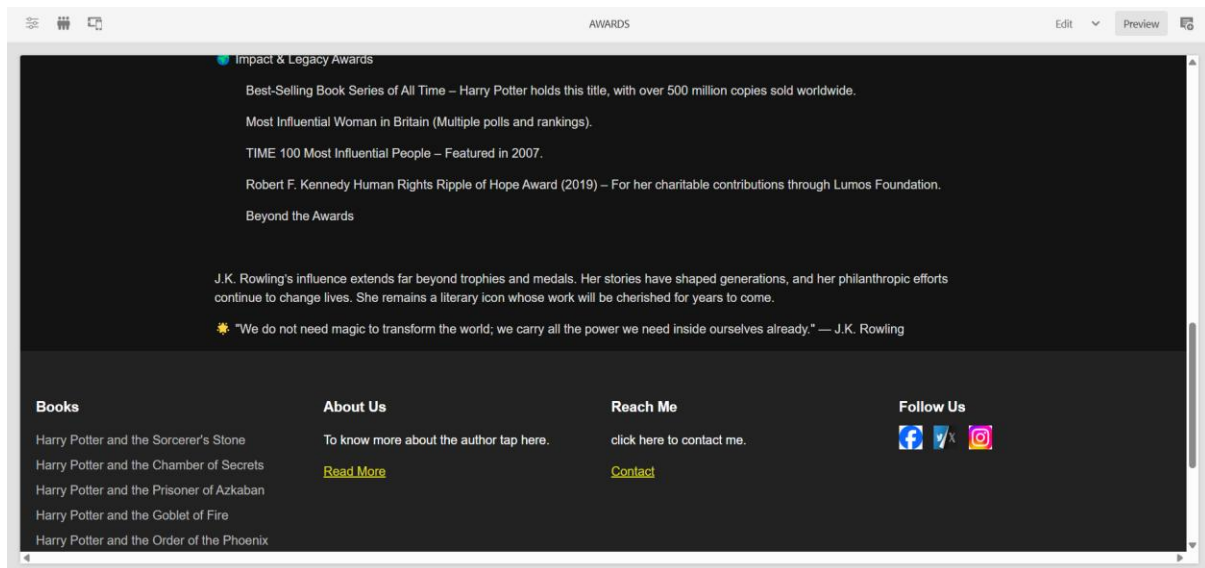




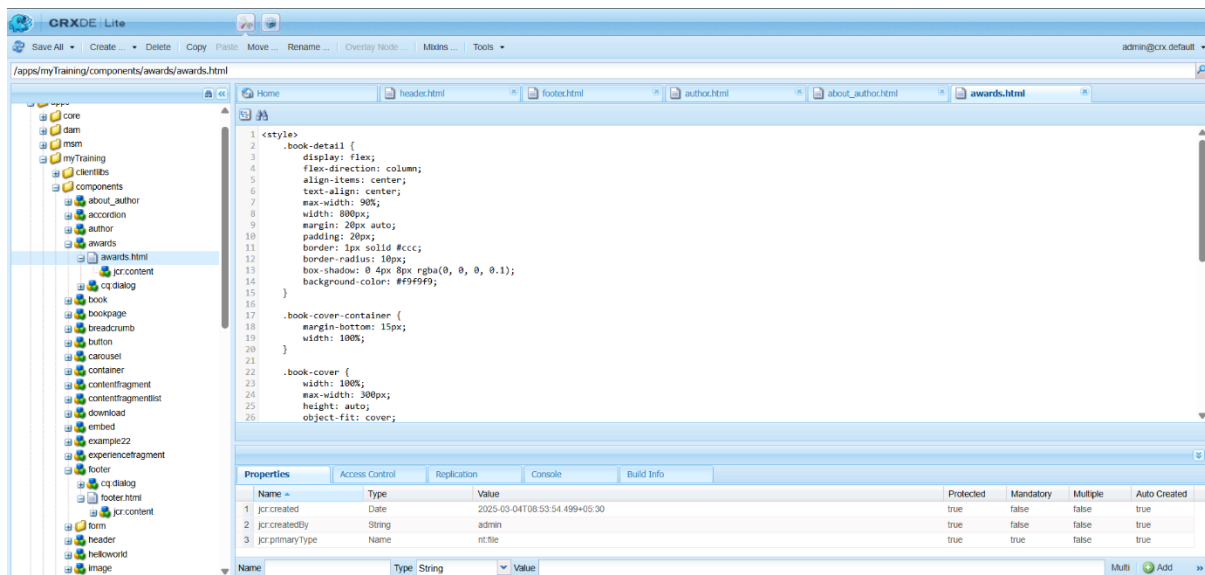
Awards page:

Created using award component and listed the awards won as text



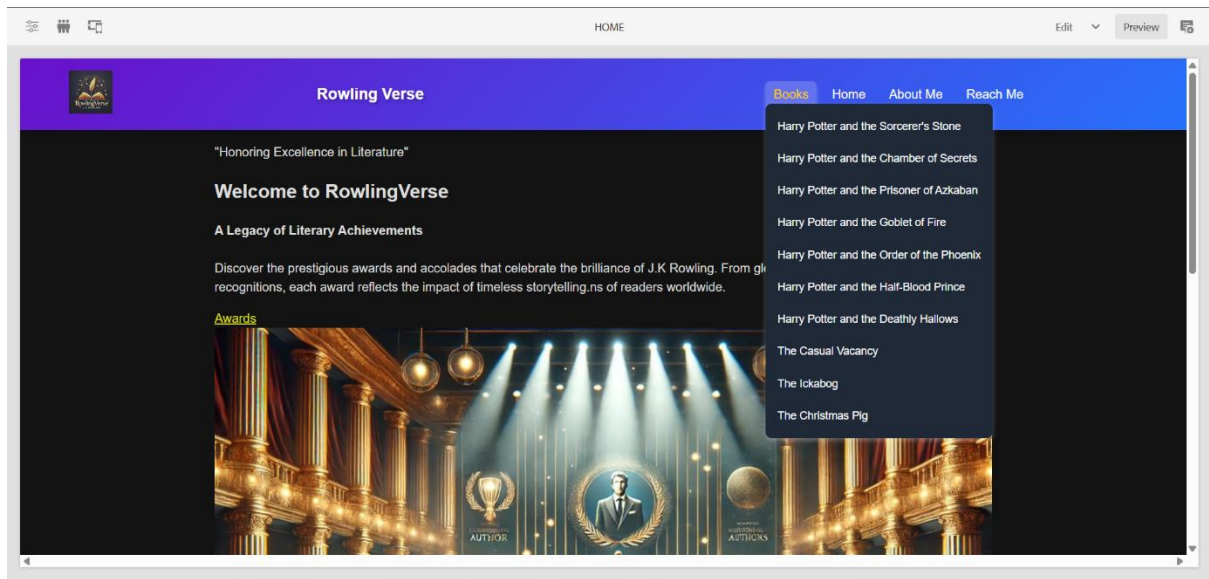


Code for awards page:

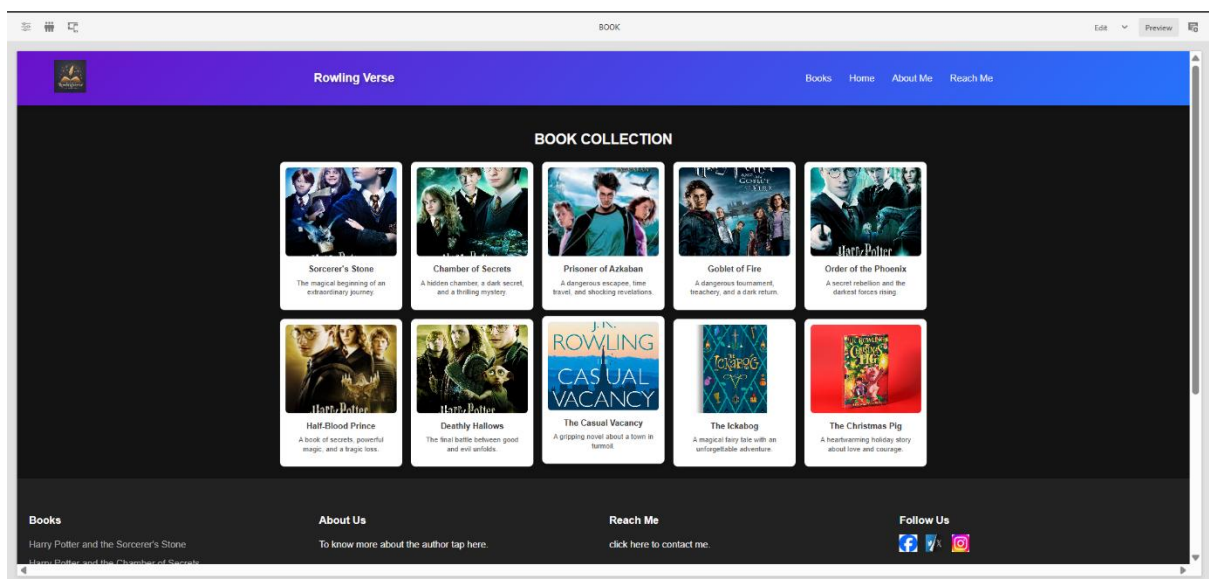


Books page:

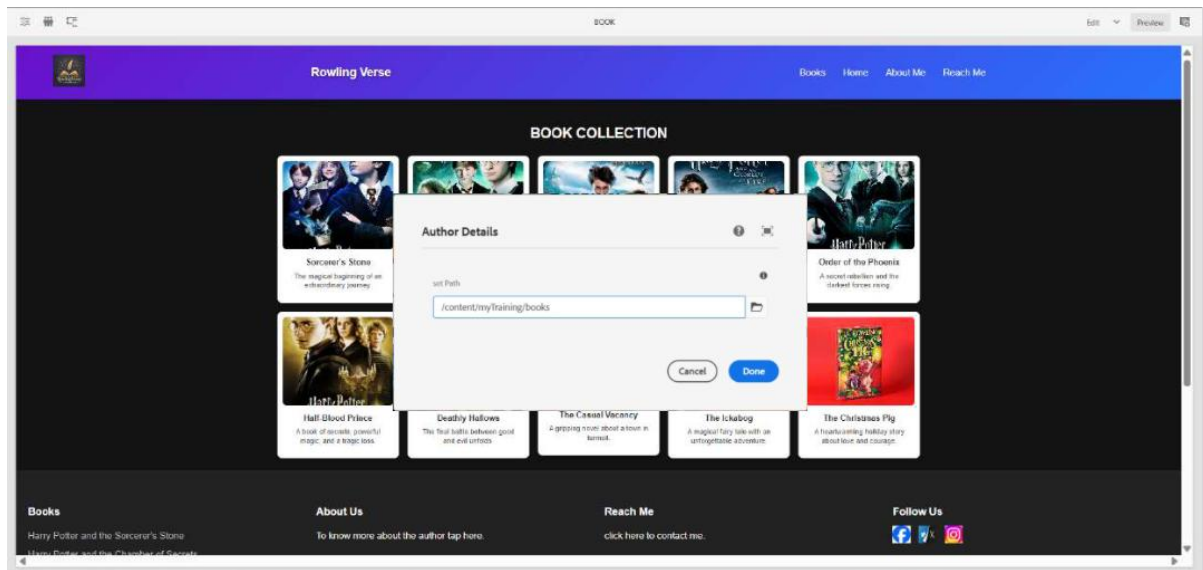
On Hover it will list the books written and on tapping it ,it will open books page.



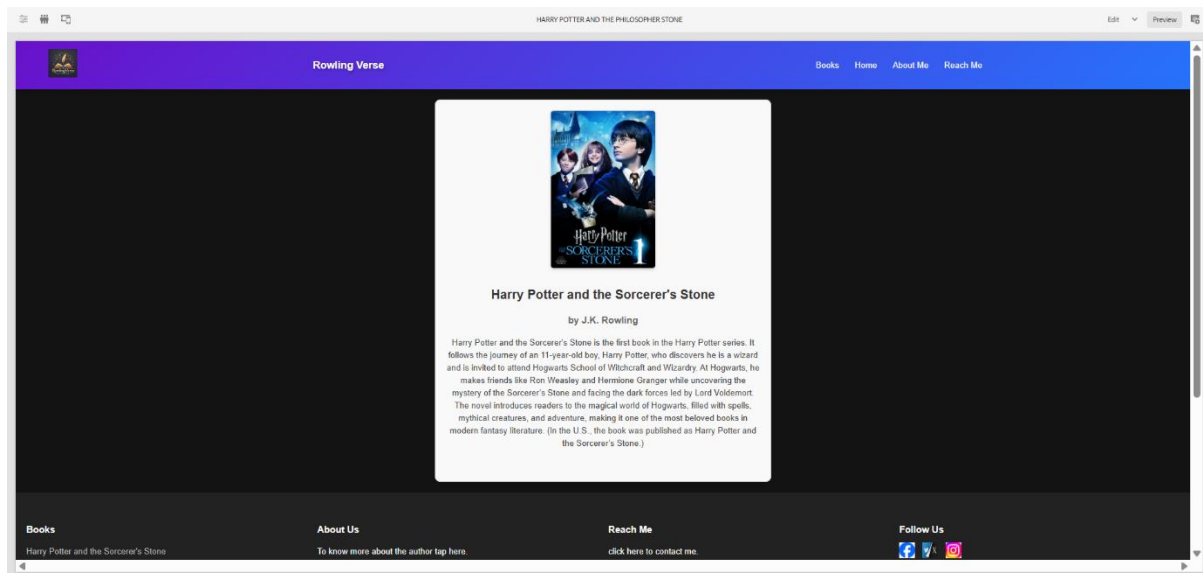
Book cards :



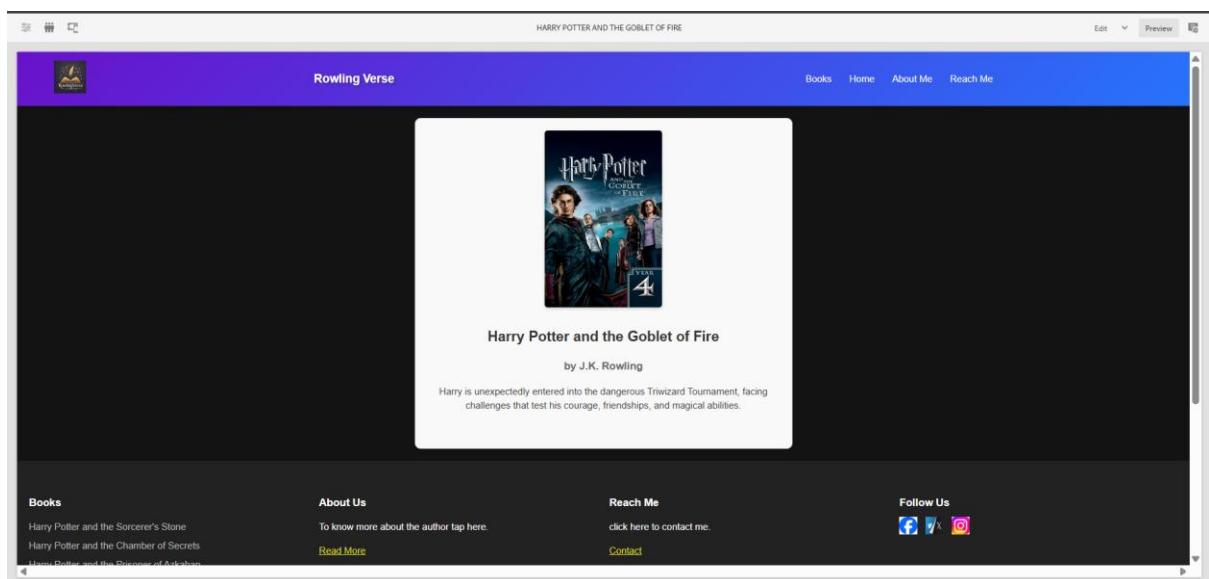
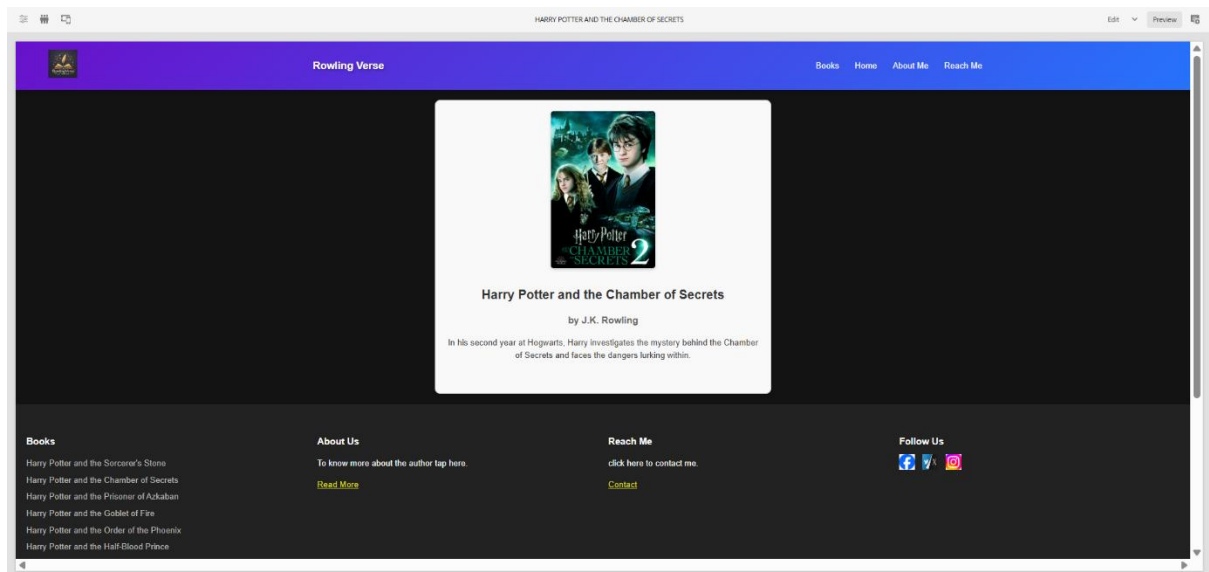
Fetching path for bookpages:



On tapping specific books in book list:

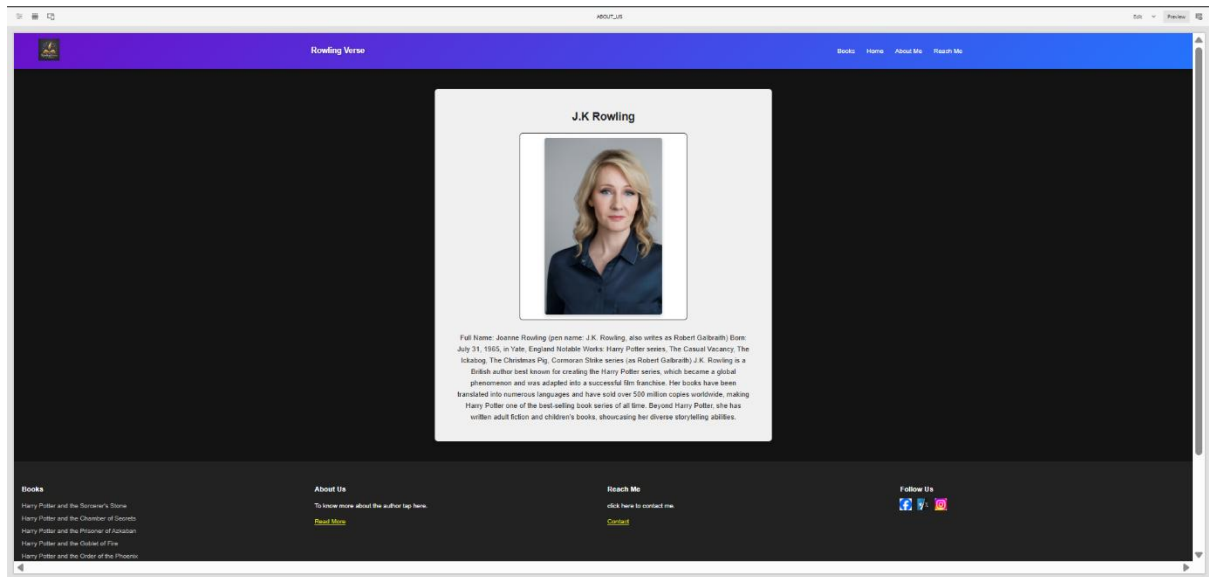


Another book:

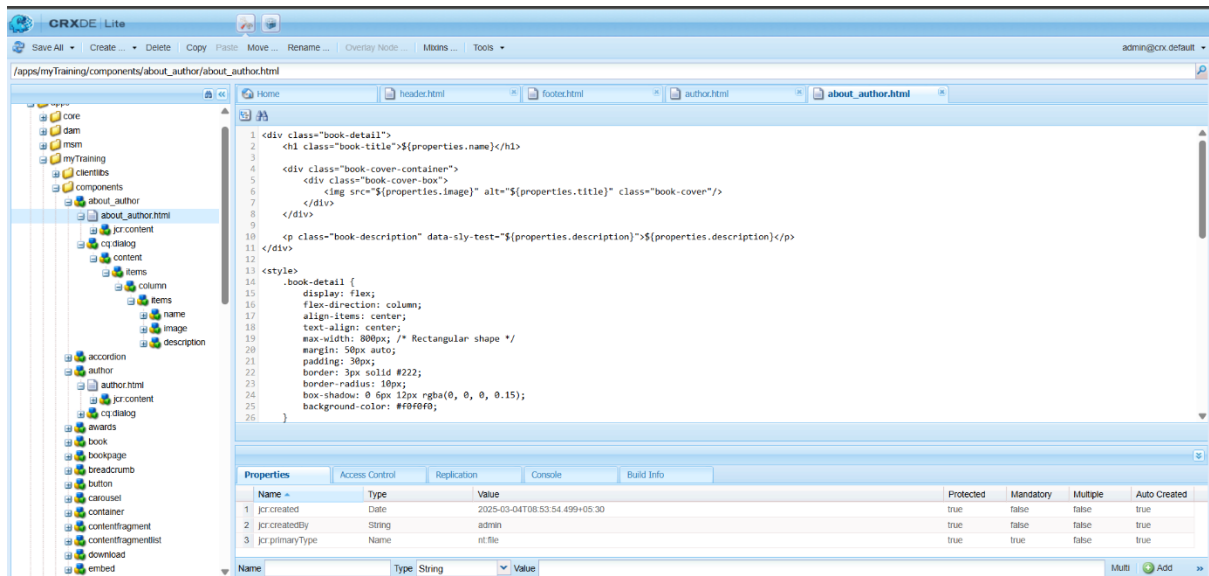


About me(Author) page:

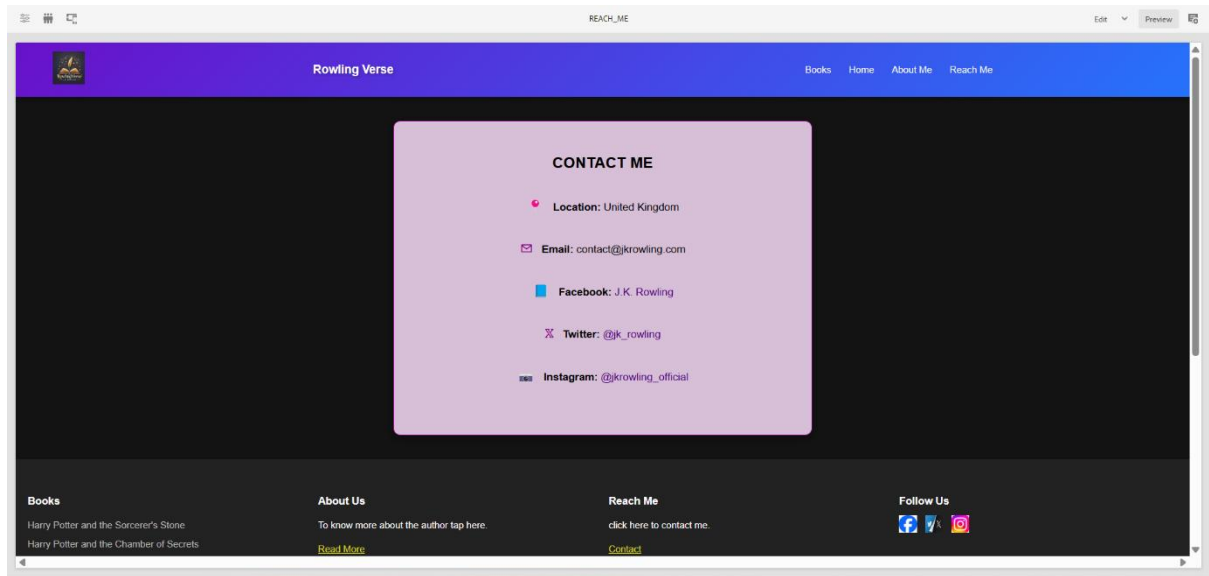
Used about_author(custom) component to create this page.



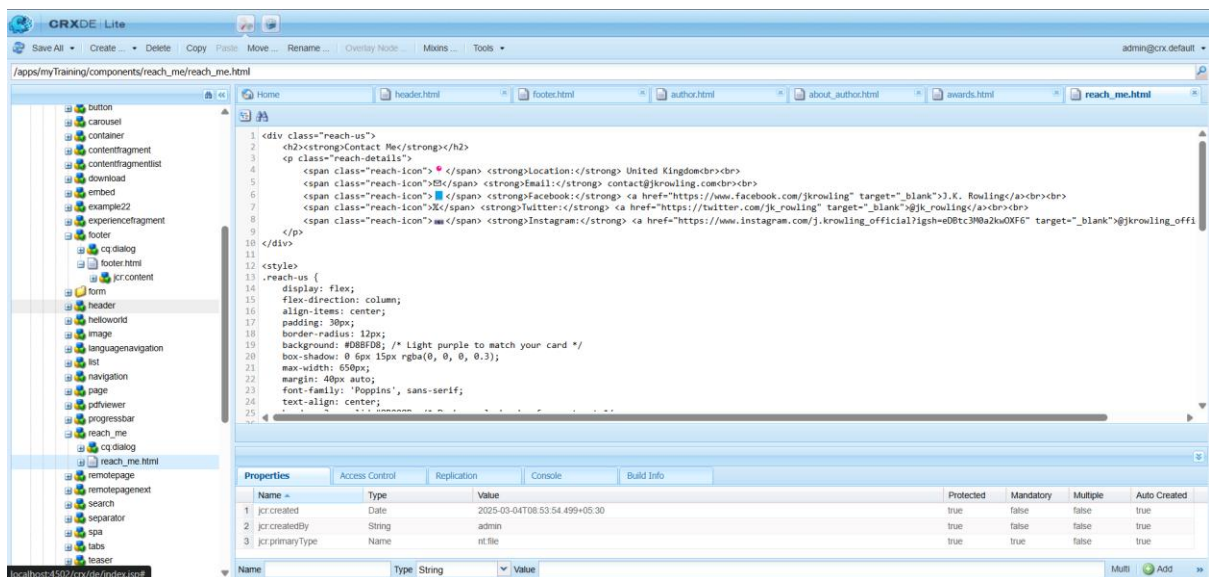
code for about author page:



Reach me page:



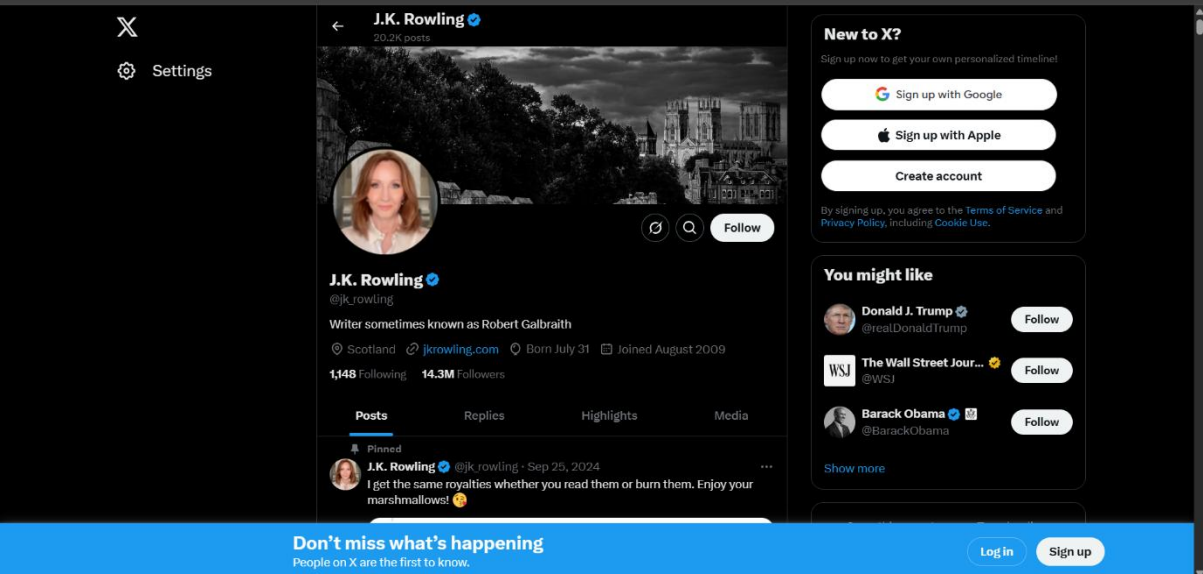
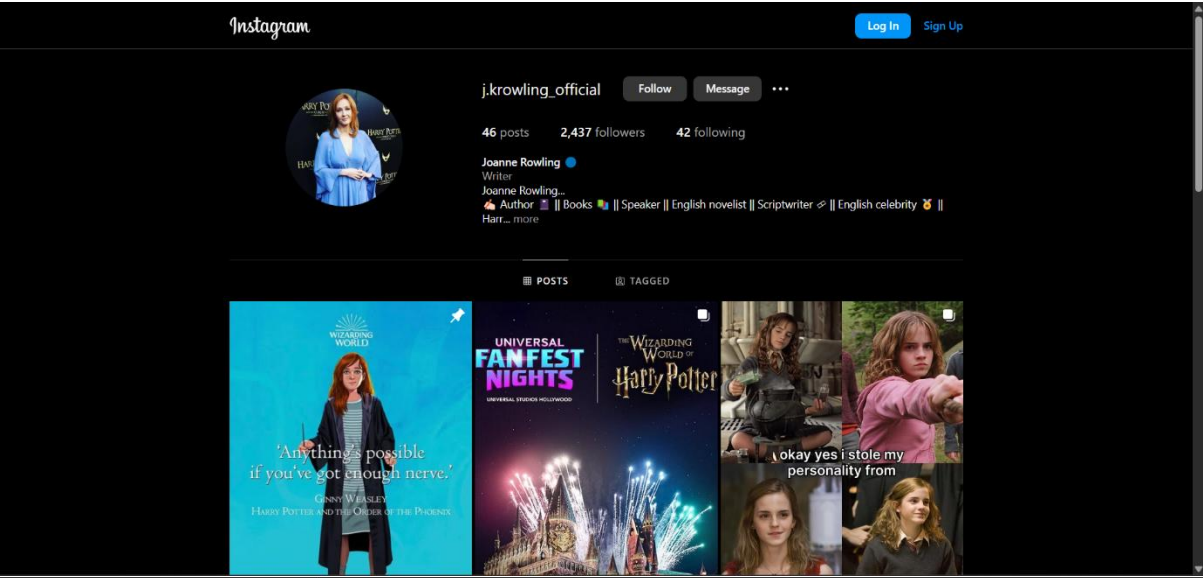
Code for reach me page:



Social media links on reach me page:



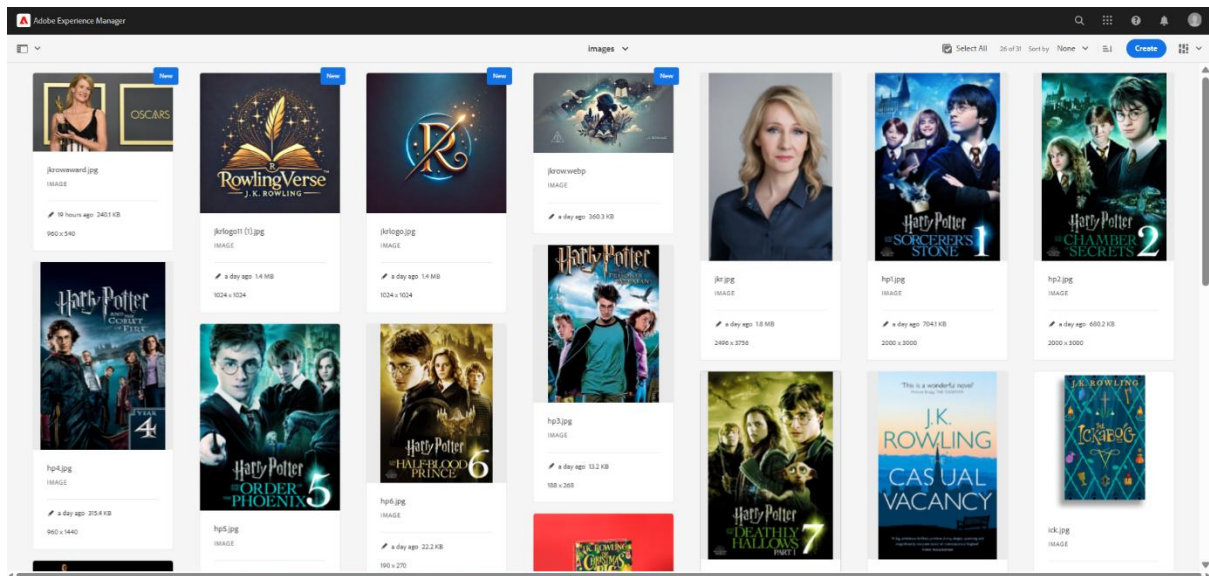
Log in or sign up for Facebook to connect with friends, family and people you know.



Don't miss what's happening
People on X are the first to know.



Assets(images used):

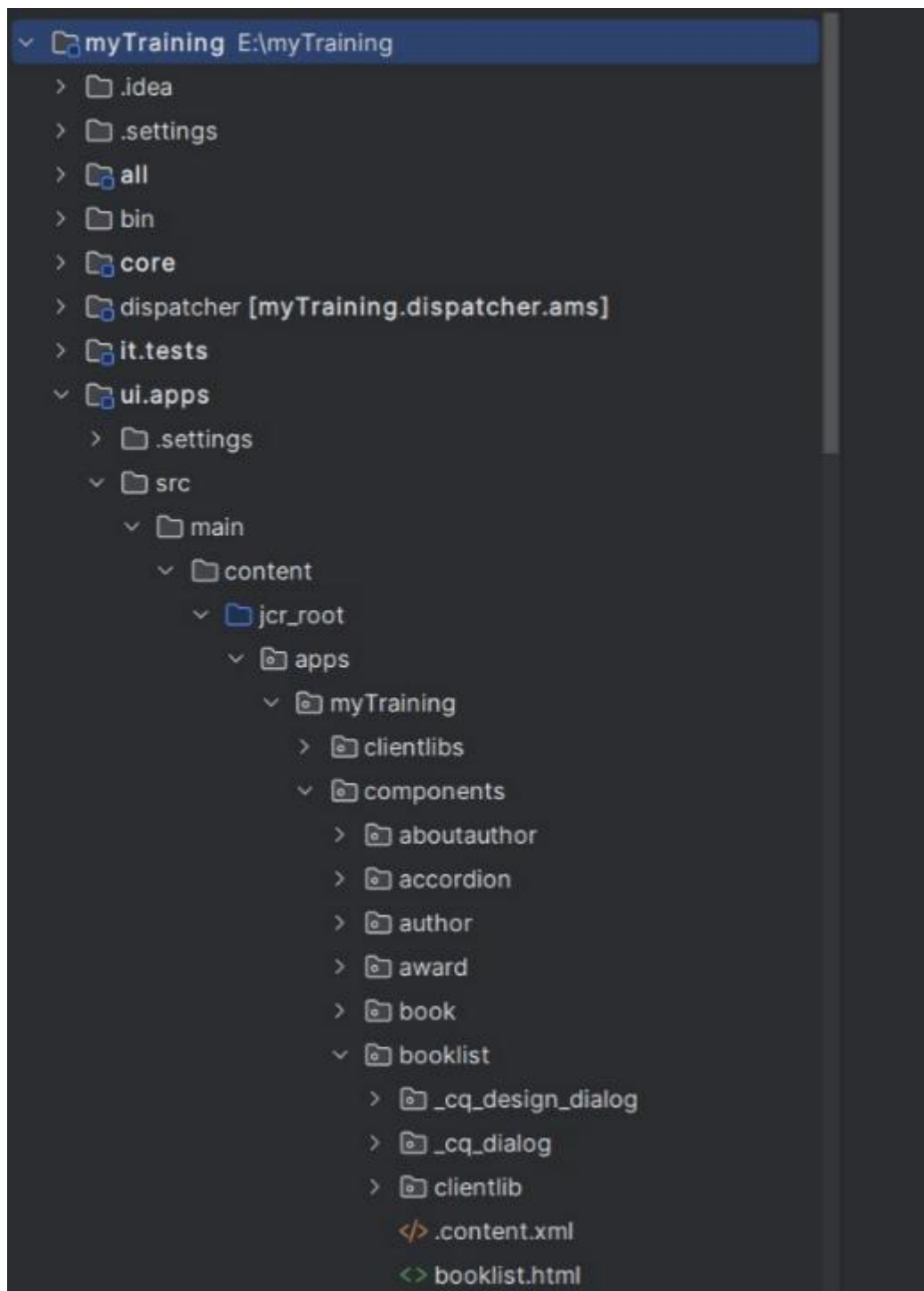


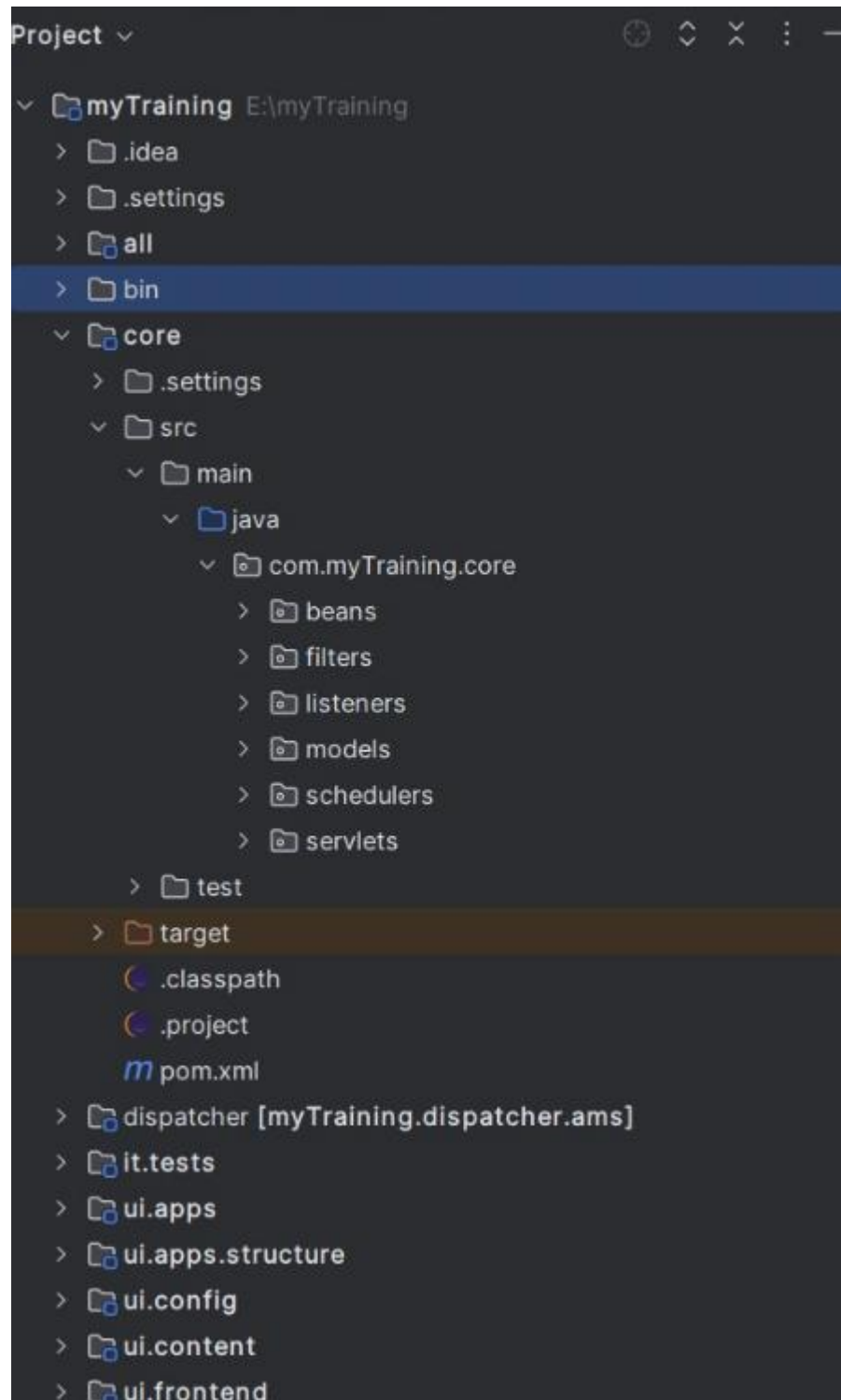
other codes:

```
[INFO] --- compiler:3.8.1:testCompile (default-testCompile) @ myTraining.ui.tests ---
[INFO] Not compiling test sources
[INFO] --- surefire:2.22.1:test (default-test) @ myTraining.ui.tests ---
[INFO] Tests are skipped.
[INFO] --- assembly:3.3.0:single (make-assembly) @ myTraining.ui.tests ---
[INFO] Reading assembly descriptor: E:\myTraining\ui.tests\assembly-ui-test-docker-context.xml
[INFO] Building tar: E:\myTraining\ui.tests\target\myTraining.ui.tests-1.0.0-SNAPSHOT-ui-test-docker-context.tar.gz
[INFO] --- install:2.5.2:install (default-install) @ myTraining.ui.tests ---
[INFO] No primary artifact to install, installing attached artifacts instead.
[INFO] Installing E:\myTraining\ui.tests\pom.xml to C:\Users\Praveena.m2\repository\com\myTraining\myTraining.ui.tests-1.0.0-SNAPSHOT.pom
[INFO] Installing E:\myTraining\ui.tests\target\myTraining.ui.tests-1.0.0-SNAPSHOT-ui-test-docker-context.tar.gz to C:\Users\Praveena.m2\repository\com\myTraining\myTraining.ui.tests-1.0.0-SNAPSHOT-ui-test-docker-context.tar.gz
[INFO] Reactor Summary for myTraining 1.0.0-SNAPSHOT:
[INFO] myTraining ..... SUCCESS [ 1.043 s]
[INFO] myTraining - Core ..... SUCCESS [ 9.585 s]
[INFO] myTraining - UI Frontend ..... SUCCESS [ 26.441 s]
[INFO] myTraining - Repository Structure Package ..... SUCCESS [ 2.518 s]
[INFO] myTraining - UI apps ..... SUCCESS [ 18.877 s]
[INFO] myTraining - UI content ..... SUCCESS [ 7.191 s]
[INFO] myTraining - UI config ..... SUCCESS [ 1.637 s]
[INFO] myTraining - All ..... SUCCESS [ 0.782 s]
[INFO] myTraining - Integration Tests ..... SUCCESS [ 21.345 s]
[INFO] myTraining - Dispatcher ..... SUCCESS [ 0.341 s]
[INFO] myTraining - UI Tests ..... SUCCESS [ 0.195 s]
[INFO] BUILD SUCCESS
[INFO] Total time: 01:34 min
[INFO] Finished at: 2025-03-14T11:57:31+05:30
[INFO]
E:\myTraining\
```

myTraining E:\myTraining

- > .idea
- > .settings
- > all
- > bin
- > core
- > dispatcher [myTraining.dispatcher.ams]
- > it.tests
- > ui.apps
- > ui.apps.structure
- > ui.config
- > ui.content
- > ui.frontend
- > ui.tests
- .classpath
- .gitattributes
- .gitignore
- .project
- archetype.properties
- LICENSE
- pom.xml
- README.md
- > External Libraries
- Scratches and Consoles





```
BookListBean.java x
1 package com.myTraining.core.beans;
2
3 public class BookListBean { 5 usages
4     private String path; 2 usages
5     private String title; 2 usages
6     private String description; 2 usages
7     private String image; // New Image Property 2 usages
8
9     public void setPath(String path) { 1 usage
10         this.path = path;
11     }
12
13     public void setTitle(String title) {
14         this.title = title;
15     }
16
17     public void setDescription(String description) {
18         this.description = description;
19     }
20
21     public void setImage(String image) { // New Setter for Image
22         this.image = image;
23     }
24
25     public String getPath() { no usages
26         return path;
27     }
28
29     public String getTitle() {
30         return title;
31     }
32 }
```

```
myTraining E:\myTraining
> .idea
> .settings
> all
> bin
> core
  > .settings
  > src
    > main
      > java
        > com.myTraining.core
          > beans
          > filters
          > listeners
          > models
            > BookListModel
            > HelloWorldModel
            > package-info.java
            > schedulers
            > servlets
          > test
          > target
            > .classpath
            > .project
            > pom.xml
          > dispatcher [myTraining.dispatcher.ams]
          > it.tests
          > ui.apps
          > ui.apps.structure
31 @Model(adaptables = Resource.class, 2 usages
32     defaultInjectionStrategy = DefaultInjectionStrategy.OPTIONAL)
33
34 public class BookListModel {
35     private static final Logger LOGGER = LoggerFactory.getLogger(BookListModel.class); 2 usages
36     List<BookListBean> bookListBeansArray=null; 3 usages
37     @Inject 2 usages
38     private String bookRootPath;
39     @Self 1 usage
40     Resource resource;
41
42
43     public String getBookRootPath() { no usages
44         return bookRootPath;
45     }
46
47     @PostConstruct no usages
48     protected void init() {
49         ResourceResolver resourceResolver = resource.getResourceResolver();
50         Session session = resourceResolver.adaptTo(Session.class);
51
52         QueryBuilder builder = resourceResolver.adaptTo(QueryBuilder.class);
53
54         Map<String, String> predicate = new HashMap<>();
55         predicate.put("path", bookRootPath);
56         predicate.put("type", "cq:Page");
57
58         Query query = null;
59
60         try {
61             query = builder.createQuery(PredicateGroup.create(predicate), session);
62             fetch(query);
63         } catch (Exception e) {
64             LOGGER.error(e.getMessage());
65         }
66     }
67 }
```