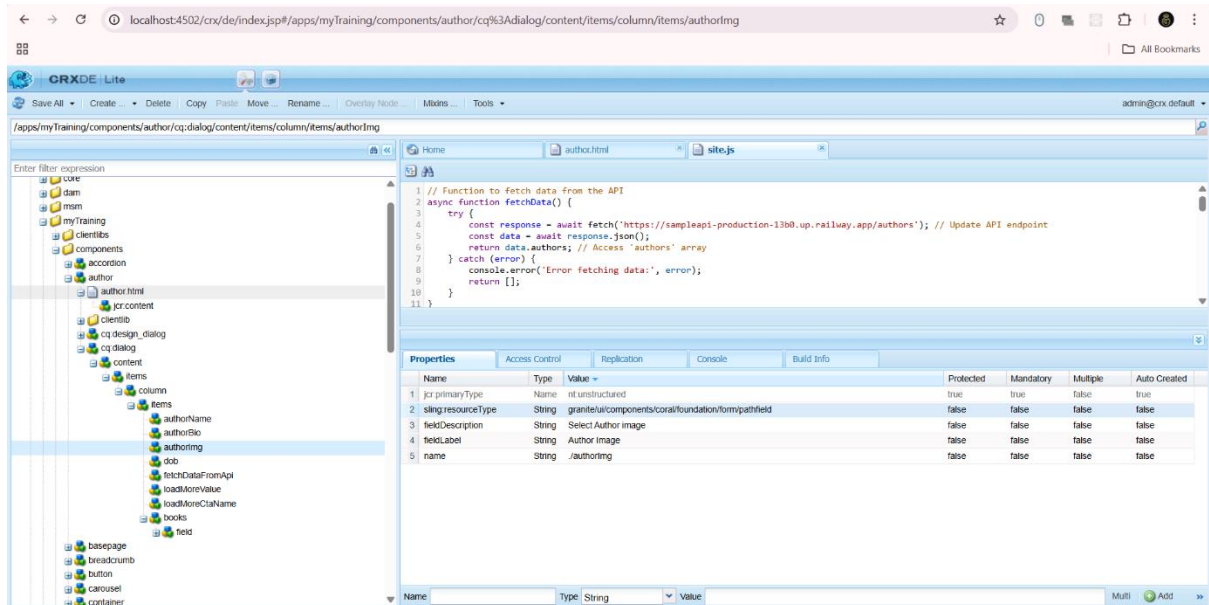
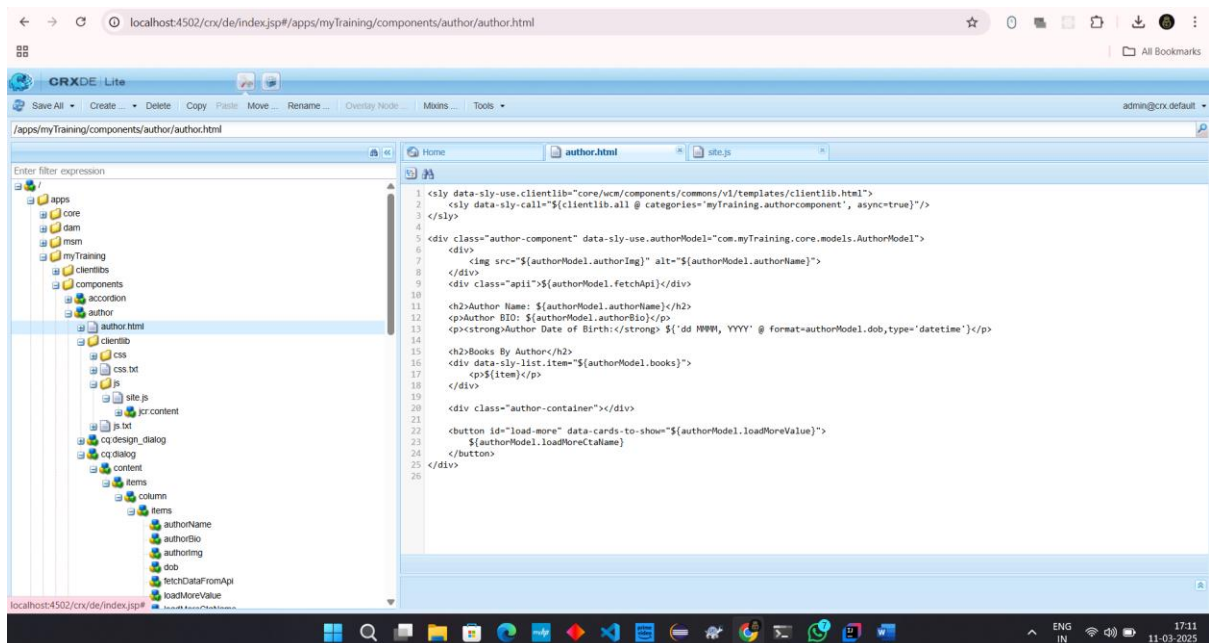


# AEM CARD CREATING TASK & FETCHING DETAILS USING API

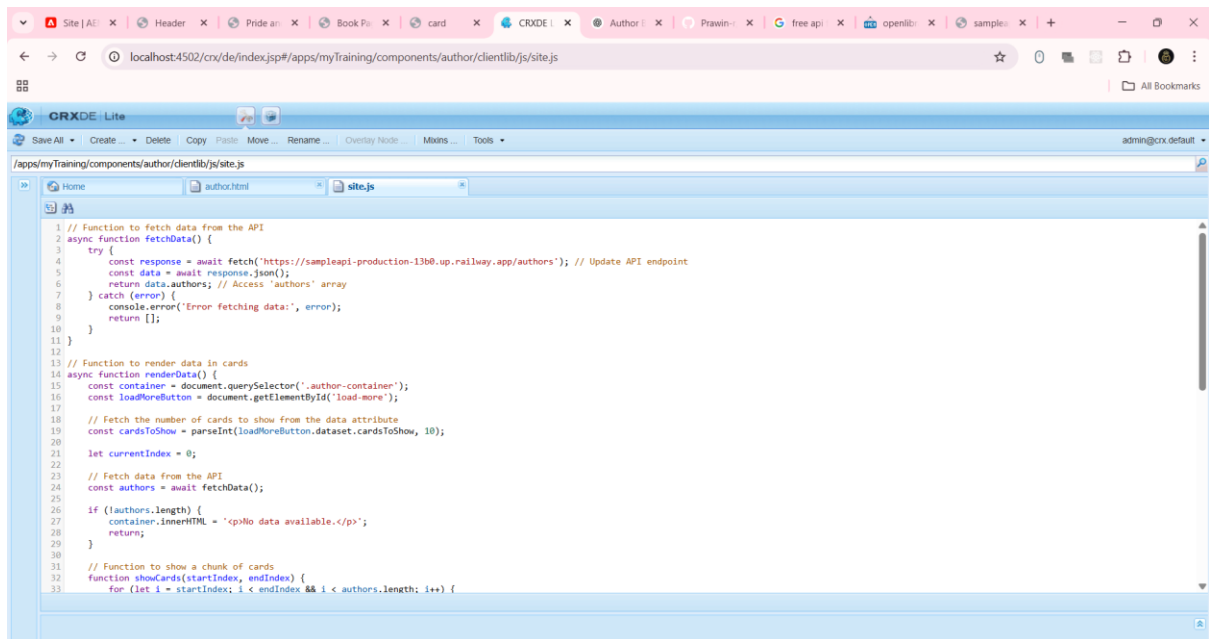
Creating author component for fetching card details:



The HTML file renders the book cards with details like image, title, description, and publish date.



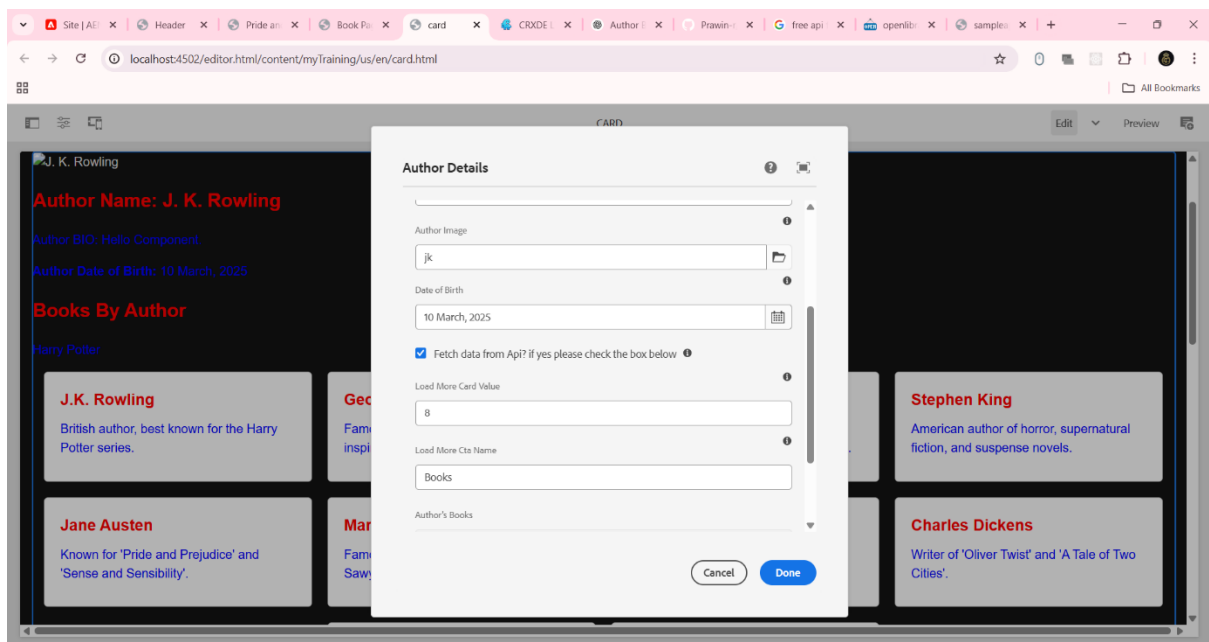
## Adding site.js in clientlibs folder for fetching data using API url



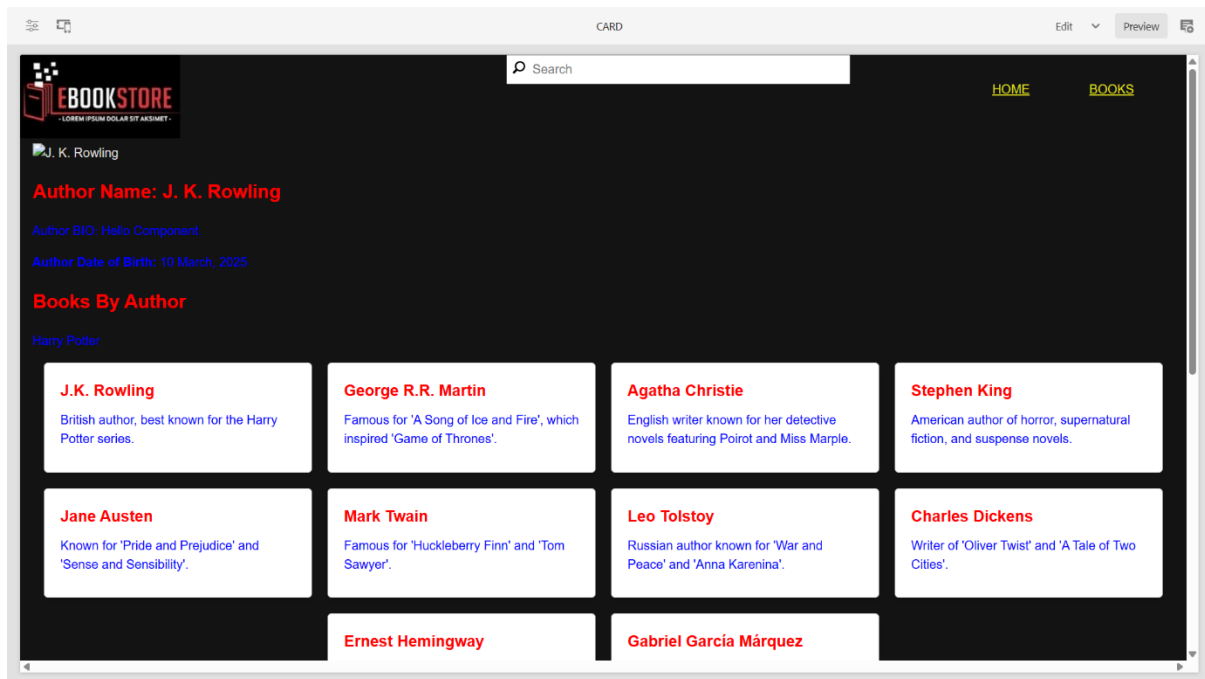
```
1 // Function to fetch data from the API
2 async function fetchData() {
3   try {
4     const response = await fetch('https://sampleapi-production-13b8.up.railway.app/authors'); // Update API endpoint
5     const data = await response.json();
6     return data.authors; // Access 'authors' array
7   } catch (error) {
8     console.error('Error fetching data:', error);
9   }
10 }
11
12
13 // Function to render data in cards
14 async function renderData() {
15   const container = document.querySelector('.author-container');
16   const loadMoreButton = document.getElementById('load-more');
17   // Fetch the number of cards to show from the data attribute
18   const cardsToShow = parseInt(loadMoreButton.dataset.cardsToShow, 10);
19   let currentIndex = 0;
20   // Fetch data from the API
21   const authors = await fetchData();
22   if (!authors.length) {
23     container.innerHTML = '<p>No data available.</p>';
24     return;
25   }
26   // Function to show a chunk of cards
27   function showCards(startIndex, endIndex) {
28     for (let i = startIndex; i < endIndex && i < authors.length; i++) {
```

Author component image:

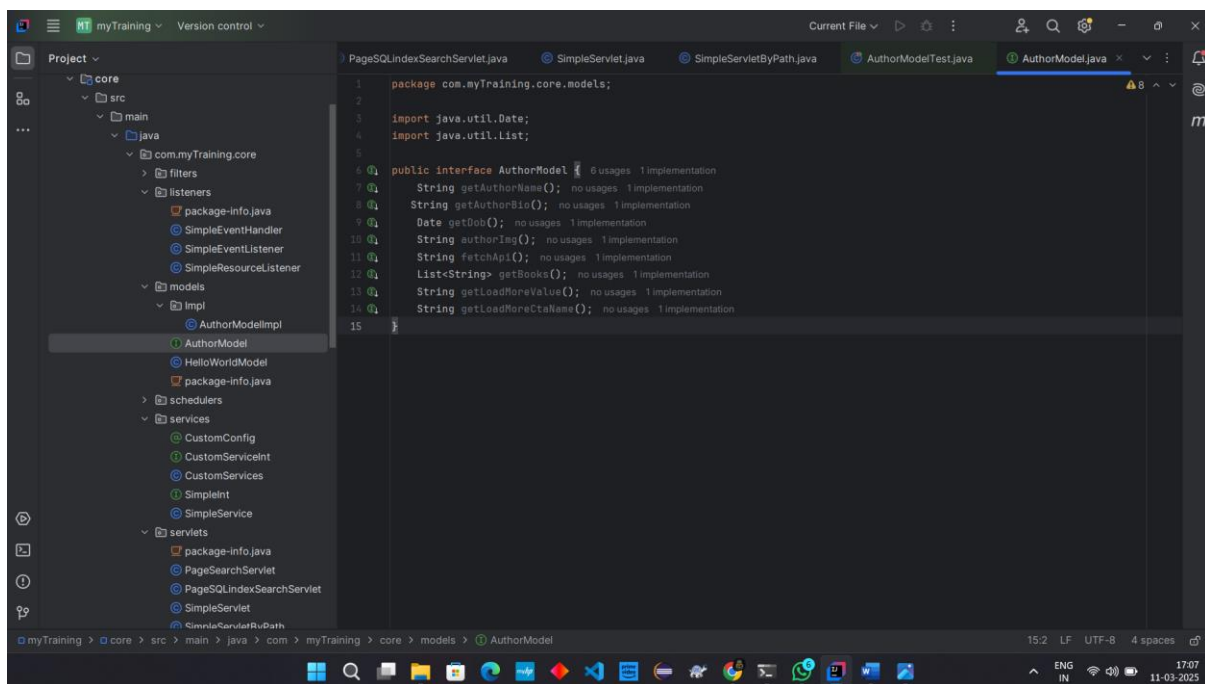
The below author component contains details of author name , author dob, and a checkbox for data fetching using Api.



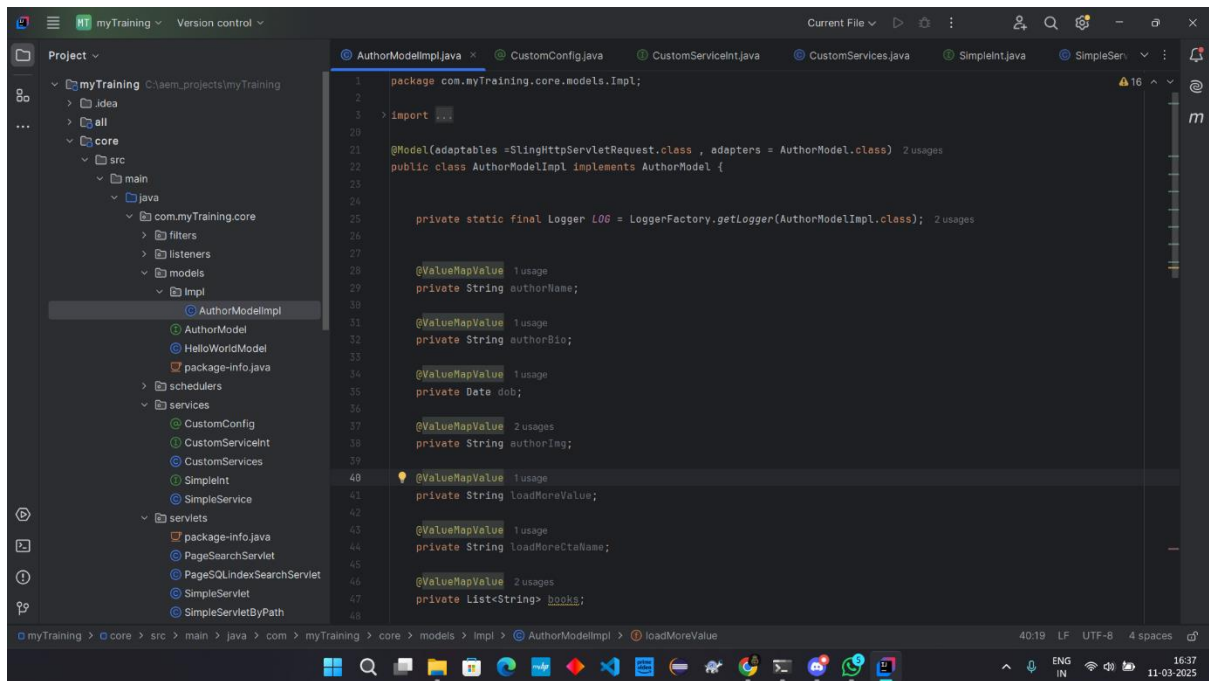
The author card details containing their name and bio .



The AuthorModel file has been modified for to get the details from Api.



This is AuthorModelImpl file modified according to AuthorModel



```
1 package com.myTraining.core.models.impl;
2
3 > import ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21 @Model(adaptables = {SlingHttpServletRequest.class, adapters = AuthorModel.class}) 2 usages
22 public class AuthorModelImpl implements AuthorModel {
23
24
25     private static final Logger LOG = LoggerFactory.getLogger(AuthorModelImpl.class); 2 usages
26
27
28     @ValueMapValue 1 usage
29     private String authorName;
30
31     @ValueMapValue 1 usage
32     private String authorBio;
33
34     @ValueMapValue 1 usage
35     private Date dob;
36
37     @ValueMapValue 2 usages
38     private String authoring;
39
40     @ValueMapValue 1 usage
41     private String loadMoreValue;
42
43     @ValueMapValue 1 usage
44     private String loadMoreEtaName;
45
46     @ValueMapValue 2 usages
47     private List<String> books;
48 }
```