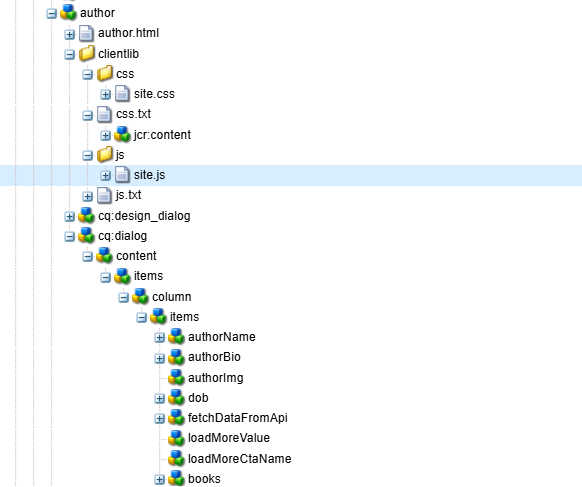
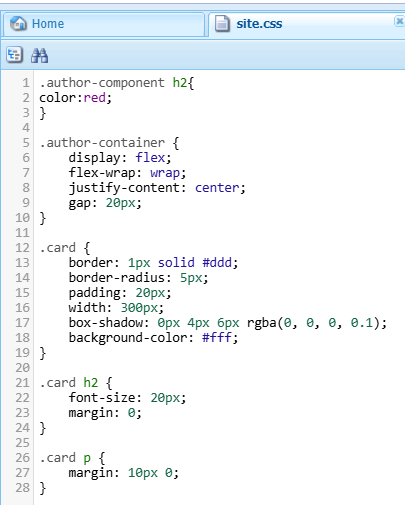
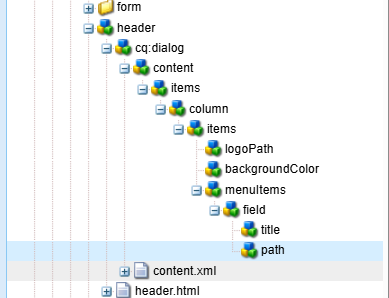
Author Component:



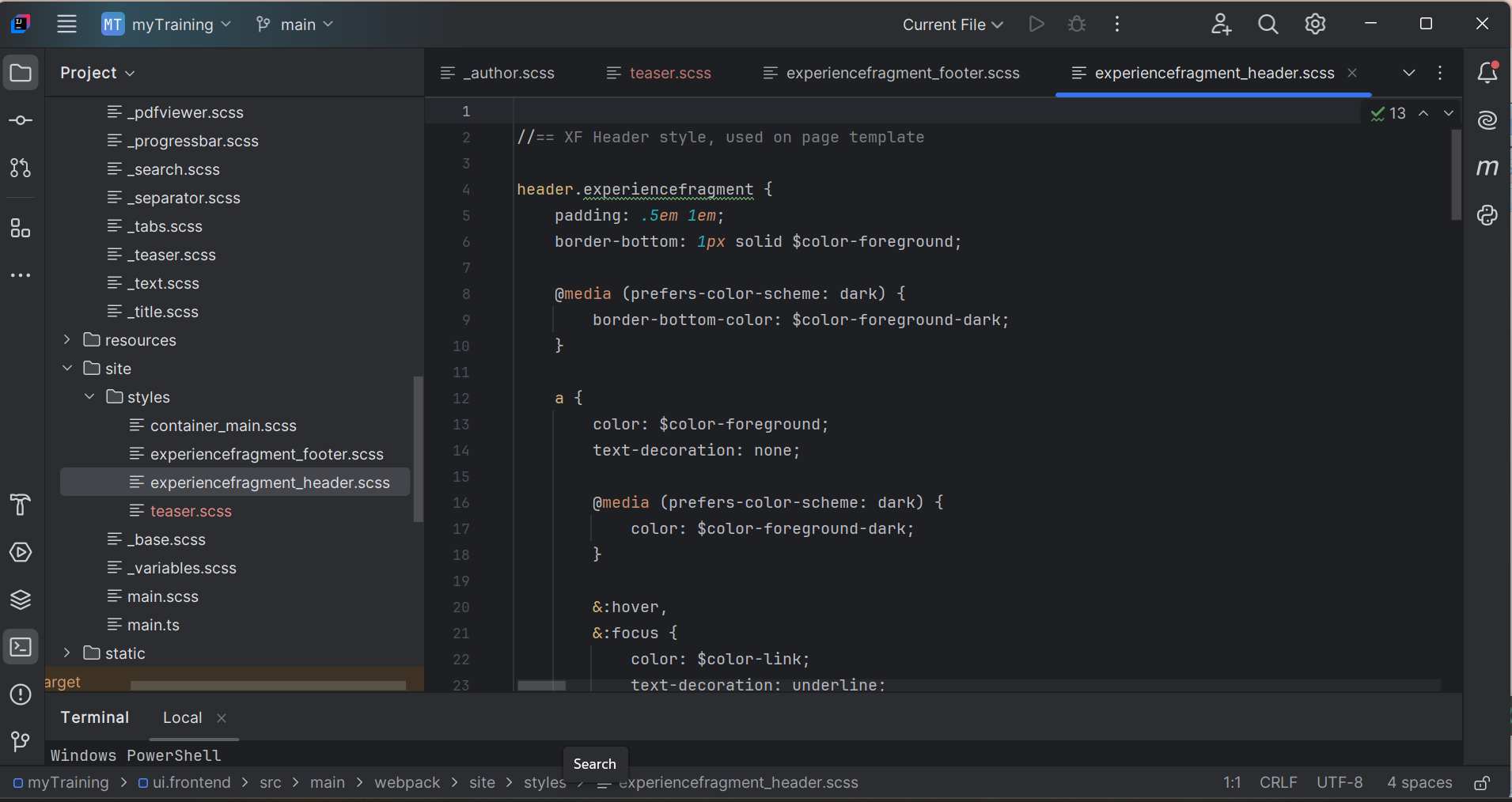
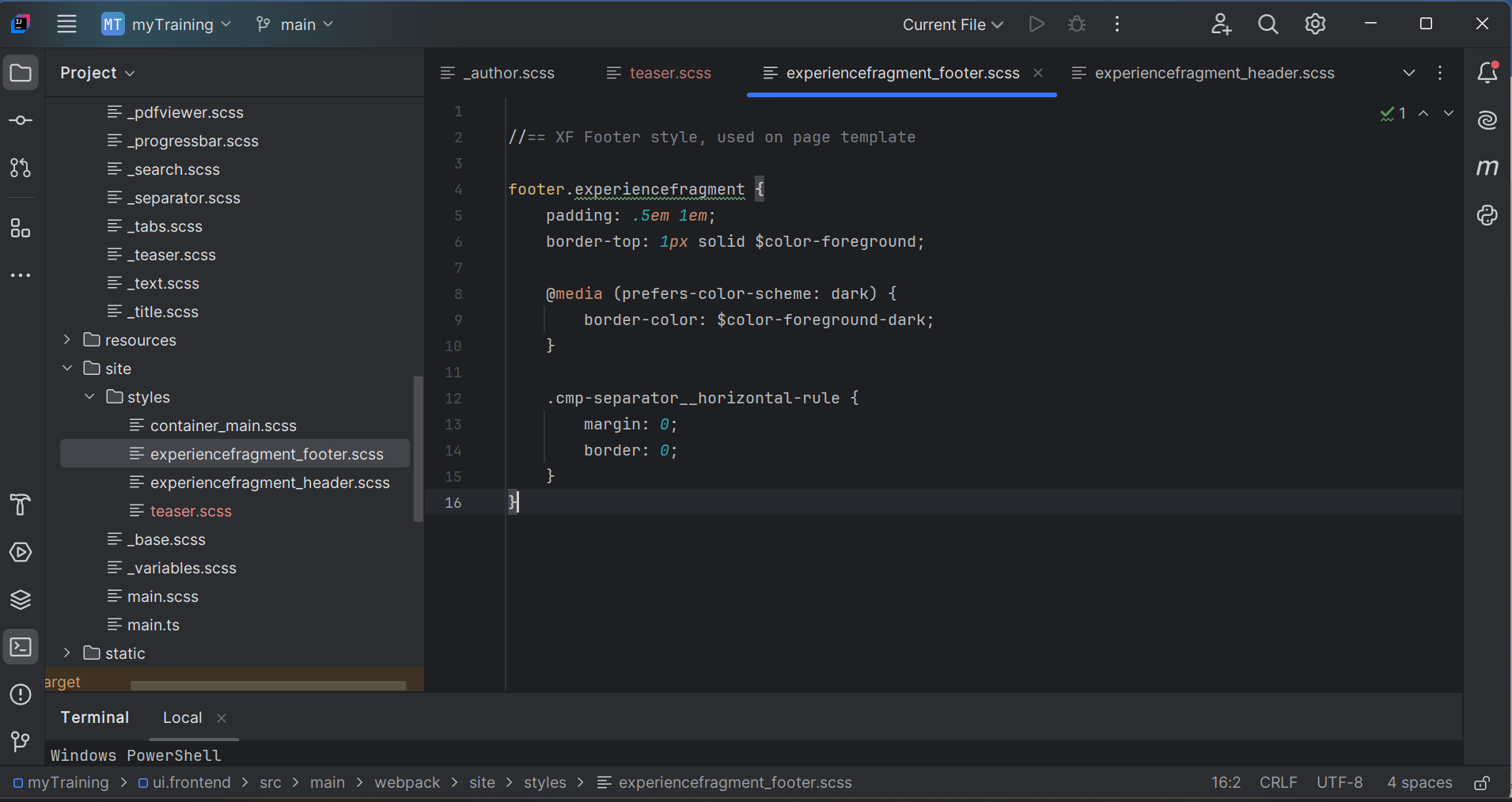
site.css:



Header & Footer:



🡪Experience frahgment setup for both header & Footer:

🡪BookListServlet:

package com.myTraining.core.servlets;

import com.day.cq.search.PredicateGroup;

import com.day.cq.search.Query;

import com.day.cq.search.QueryBuilder;

import com.day.cq.search.result.Hit;

import com.day.cq.search.result.SearchResult;

import org.apache.sling.api.SlingHttpServletRequest;

import org.apache.sling.api.SlingHttpServletResponse;

import org.apache.sling.api.servlets.SlingAllMethodsServlet;

import org.apache.sling.servlets.annotations.SlingServletPaths;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import org.osgi.service.component.annotations.Component;

import org.osgi.service.component.annotations.Reference;

import javax.jcr.RepositoryException;

import javax.jcr.Session;

import javax.servlet.Servlet;

import javax.servlet.ServletException;

import java.io.IOException;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

@Component(service = Servlet.class)

@SlingServletPaths("/bin/training/searchpages")

public class BookListServlet extends SlingAllMethodsServlet {

@Reference

private QueryBuilder queryBuilder;

@Override

protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response) throws ServletException, IOException {

response.setContentType("application/json");

response.setCharacterEncoding("UTF-8");

JSONObject jsonResponse = new JSONObject();

JSONArray resultsArray = new JSONArray();

Session session = request.getResourceResolver().adaptTo(Session.class);

if (session == null) {

try {

jsonResponse.put("error", "JCR Session is null");

} catch (JSONException e) {

throw new RuntimeException(e);

}

response.getWriter().write(jsonResponse.toString());

return;

}

try {

// Query Predicate Map

Map<String, String> predicateMap = new HashMap<>();

predicateMap.put("path", "/content/myTraining/books"); // Search under this path

predicateMap.put("type", "cq:Page"); // Search only cq:Page nodes

predicateMap.put("p.limit", "-1"); // Fetch all results

Query query = queryBuilder.createQuery(PredicateGroup.create(predicateMap), session);

SearchResult searchResult = query.getResult();

List<Hit> hits = searchResult.getHits();

for (Hit hit : hits) {

JSONObject pageObject = new JSONObject();

pageObject.put("path", hit.getPath());

try {

// Fetch `jcr:title` from `jcr:content`

String title = hit.getProperties().get("jcr:title", "");

pageObject.put("title", title);

// Fetch `jcr:description` from `jcr:content`

String description = hit.getProperties().get("jcr:description", "");

pageObject.put("description", description);

} catch (RepositoryException e) {

pageObject.put("title", "N/A"); // Default title if not found

pageObject.put("description", "N/A"); // Default description if not found

}

resultsArray.put(pageObject);

}

jsonResponse.put("results", resultsArray);

} catch (Exception e) {

try {

jsonResponse.put("error", "Search failed: " + e.getMessage());

} catch (JSONException ex) {

throw new RuntimeException(ex);

}

}

response.getWriter().write(jsonResponse.toString());

}

}

Style for book cards to fetch api:  
site.js:  
  
// Function to fetch data from the API

async function fetchData() {

try {

const response = await fetch('https://potterapi-fedeperin.vercel.app/en/books');

const data = await response.json();

return data;

} catch (error) {

console.error('Error fetching data:', error);

return [];

}

}

// Function to render data in cards

async function renderData() {

const container = document.querySelector('.author-container');

const loadMoreButton = document.getElementById('load-more');

if (!container || !loadMoreButton) {

console.error('Required elements not found!');

return;

}

const cardsToShow = parseInt(loadMoreButton.dataset.cardsToShow, 10) || 3;

let currentIndex = 0;

const data = await fetchData();

if (!data.length) {

console.error('No books found!');

return;

}

// Function to show a chunk of cards

function showCards(startIndex, endIndex) {

for (let i = startIndex; i < endIndex && i < data.length; i++) {

const book = data[i];

const card = document.createElement('div');

card.classList.add('card');

const img = document.createElement('img');

img.src = book.cover;

img.alt = book.title;

const title = document.createElement('h2');

title.textContent = book.title;

const description = document.createElement('p');

description.textContent = book.description;

card.appendChild(img);

card.appendChild(title);

card.appendChild(description);

container.appendChild(card);

}

}

// Initially show the first few cards

showCards(currentIndex, currentIndex + cardsToShow);

currentIndex += cardsToShow;

// Load more cards when the button is clicked

loadMoreButton.addEventListener('click', function () {

showCards(currentIndex, currentIndex + cardsToShow);

currentIndex += cardsToShow;

if (currentIndex >= data.length) {

loadMoreButton.style.display = 'none';

}

});

if (currentIndex >= data.length) {

loadMoreButton.style.display = 'none';

}

}

// Call the renderData function

renderData();  
Other CSS:  
body, html {

background: #ececec;

color: #202020;

font-family: Helvetica Neue, Helvetica, Arial, sans-serif;

font-size: 1pc;

line-height: 1.5;

margin: 0;

}

@media (prefers-color-scheme: dark) {

body, html {

background: #131313;

color: #dfdfdf;

}

}

a {

color: #2020e0;

}

@media (prefers-color-scheme: dark) {

a {

color: #dfdf1f;

}

}

button, input, optgroup, select, textarea {

font: inherit;

}

.author-component p {

color: blue;

}

.cmp-test123 {

color: red;

}

.cmp-helloworld\_\_item-label {

margin-bottom: 0;

}

.cmp-helloworld\_\_item-output {

margin-top: 0;

}

.cmp-list\_\_item-title {

color: red;

}

.cmp-navigation\_\_item--level-0:hover > .cmp-navigation\_\_group {

background: #ececec;

}

main.container {

padding: 0.5em 1em;

}

footer.experiencefragment {

border-top: 1px solid #202020;

padding: 0.5em 1em;

}

@media (prefers-color-scheme: dark) {

footer.experiencefragment {

border-color: #dfdfdf;

}

}

footer.experiencefragment .cmp-separator\_\_horizontal-rule {

border: 0;

margin: 0;

}

header.experiencefragment {

border-bottom: 1px solid #202020;

padding: 0.5em 1em;

}

@media (prefers-color-scheme: dark) {

header.experiencefragment {

border-bottom-color: #dfdfdf;

}

}

header.experiencefragment a {

color: #202020;

text-decoration: none;

}

@media (prefers-color-scheme: dark) {

header.experiencefragment a {

color: #dfdfdf;

}

}

header.experiencefragment a:focus,

header.experiencefragment a:hover {

color: #2020e0;

text-decoration: underline;

}

@media (prefers-color-scheme: dark) {

header.experiencefragment a:focus,

header.experiencefragment a:hover {

color: #dfdf1f;

}

}

header.experiencefragment .cmp-container {

display: grid;

grid-template-columns: 4fr 1fr 1fr;

}

header.experiencefragment .cmp-navigation\_\_group {

list-style: none;

margin: 0;

padding: 0;

}

header.experiencefragment .cmp-navigation\_\_item--level-0 {

display: grid;

grid-template-columns: 1fr 3fr;

}

header.experiencefragment .cmp-navigation\_\_item--level-0 > .cmp-navigation\_\_group {

display: grid;

grid-template-columns: repeat(3, 1fr);

}

header.experiencefragment .cmp-navigation\_\_item-link {

display: block;

}

header.experiencefragment .cmp-navigation\_\_item--active > .cmp-navigation\_\_item-link {

font-weight: 700;

}

header.experiencefragment .cmp-languagenavigation {

position: relative;

}

header.experiencefragment .cmp-languagenavigation:before {

background-image: url('data:image/svg+xml;utf8,<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path fill="rgb(32,32,32)" d="M12.1,0.2h-0.1c-3.1,0-6.1,1.3-8.3,3.5C1.4,6,0.2,8.9,0.2,12.1c0,3.1,1.3,6.1,3.5,8.3c2.2,2.2,5.1,3.4,8.3,3.4 c0,0,0,0,0.1,0c6.5,0,11.7-5.3,11.7-11.8C23.7,5.5,18.5,0.3,12.1,0.2z"/></svg>');

background-repeat: no-repeat;

content: " ";

display: block;

height: 1.5em;

}

@media (prefers-color-scheme: dark) {

header.experiencefragment .cmp-languagenavigation:before {

background-image: url('data:image/svg+xml;utf8,<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path fill="rgb(223,223,223)" d="M12.1,0.2h-0.1c-3.1,0-6.1,1.3-8.3,3.5C1.4,6,0.2,8.9,0.2,12.1c0,3.1,1.3,6.1,3.5,8.3c2.2,2.2,5.1,3.4,8.3,3.4 c0,0,0,0,0.1,0c6.5,0,11.7-5.3,11.7-11.8C23.7,5.5,18.5,0.3,12.1,0.2z"/></svg>');

}

}