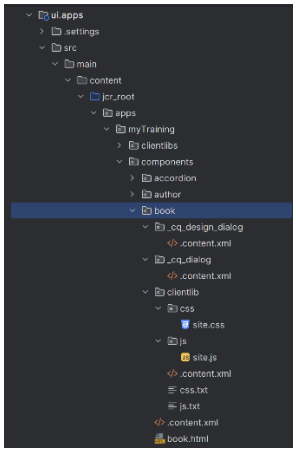
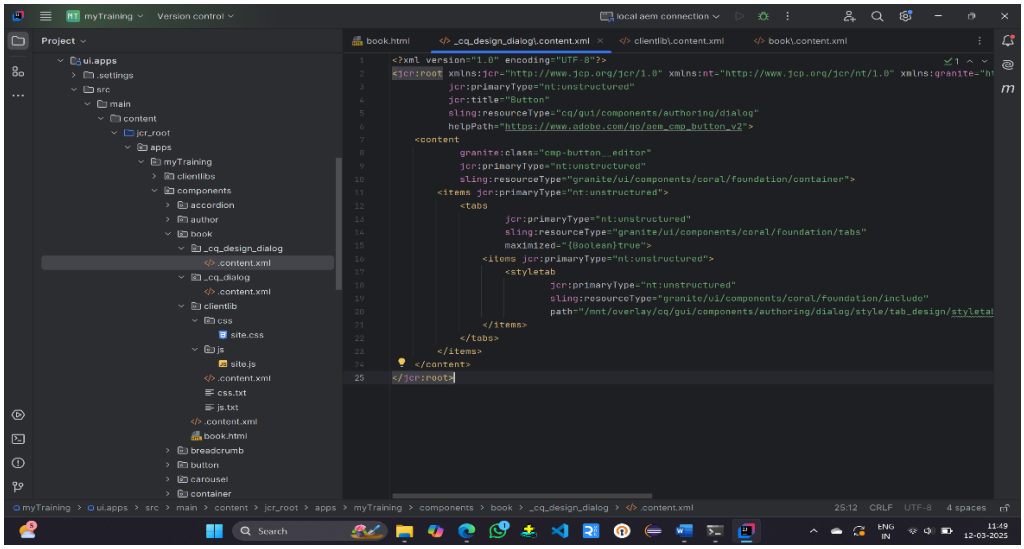
Book Component Implementation :

Components -> create a component named book and create the following folder structure



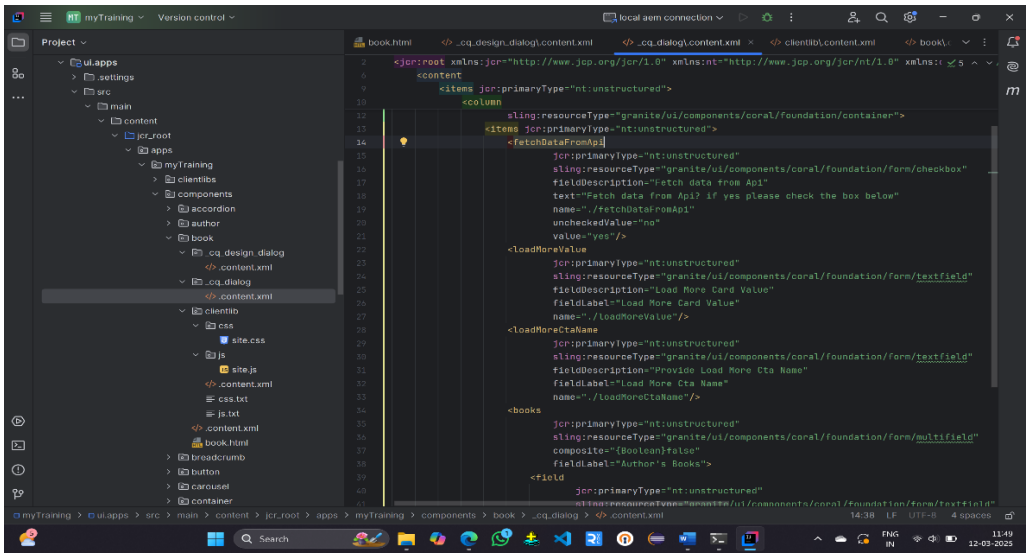
_cq_design_dialog .content.xml ->

add the following code



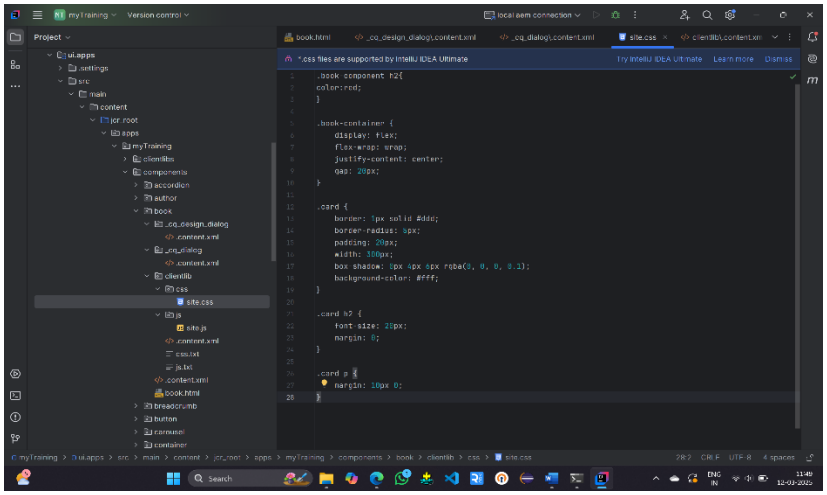
_cq_dialog .content.xml ->

add the following code



clientlib -> add the following code

Site.css

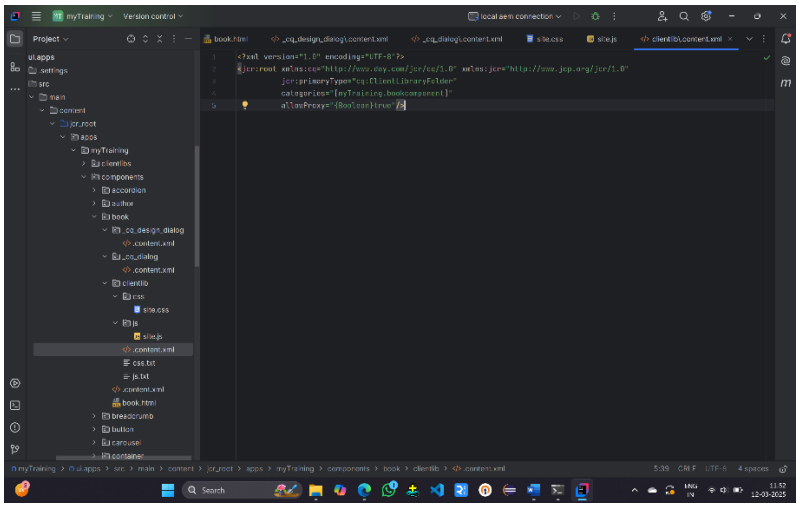


Site.js

```
// Function to fetch data from the API

async function fetchData() {
    try {
        const response = await fetch('https://jsonplaceholder.typicode.com/posts');
        const data = await response.json();
        console.log(data);
        return data;
    } catch (error) {
        console.error('Error fetching data:', error);
    }
}

// Function to render data in cards
async function renderData() {
    const container = document.querySelector('.book-container');
    const loadMoreButton = document.getElementById('load-more');

    // Fetch the number of cards to show from the data attribute
    const cardsToShow = parseInt(loadMoreButton.dataset.cardsToShow, 10); // Use the valu

    let currentIndex = 0;

    // Fetch data from the API
    const data = await fetchData();

    if (!data) {
        return;
    }
```

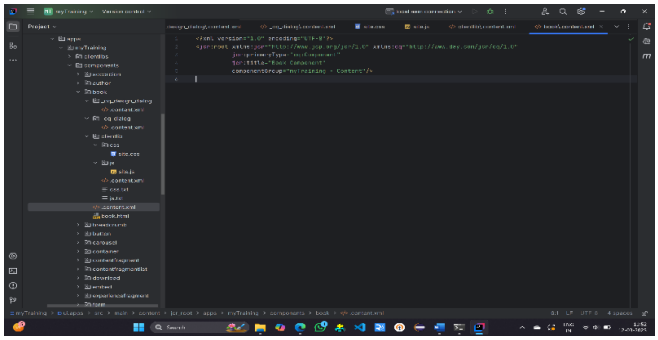
```
    // Function to show a chunk of cards
    function showCards(startIndex, endIndex) {
        for (let i = startIndex; i < endIndex && i < data.length; i++) {
            const item = data[i];
            const card = document.createElement('div');
            card.classList.add('card');

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

            const body = document.createElement('p');
            body.textContent = item.body;

            card.appendChild(title);
            card.appendChild(body);
            container.appendChild(card);
        }
    }

    // Initially show the first 'cardsToShow' cards (including the default ones)
    showCards(currentIndex, currentIndex + cardsToShow);
    currentIndex += cardsToShow;

    // Load more cards when the button is clicked
    loadMoreButton.addEventListener('click', function () {
        showCards(currentIndex, currentIndex + cardsToShow);
        currentIndex += cardsToShow;

        // Hide the "Load More" button if all cards are loaded
        if (currentIndex >= data.length) {
            loadMoreButton.style.display = 'none';
        }
    });
```

.content.xml



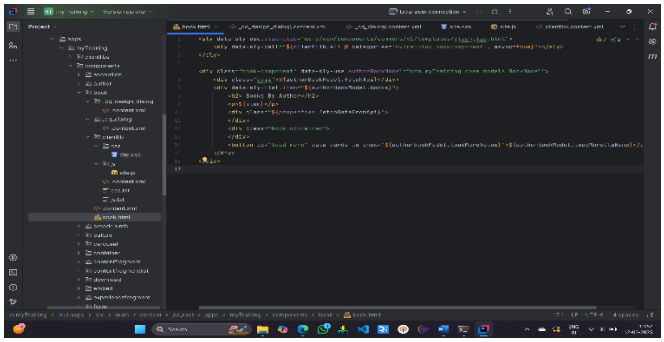
book .content.html ->

add the following code



book.html ->

add the following code



com.myTraining.core modals ->

write the following code

BookModal.java

BookModel.java | BookModelImpl.java | CustomConfig.java | CustomService.java | CustomServiceInt.java

```java
package com.myTraining.core.models;

import java.util.Date;
import java.util.List;

public interface BookModel {  4 usages  1 implementation
    String fetchApi();  no usages  1 implementation
    List<String> getBooks();  no usages  1 implementation
    String getLoadMoreValue();  no usages  1 implementation
    String getLoadMoreCtaName();  no usages  1 implementation
}
```

BookModelImp.java

BookModel.java BookModelImpl.java CustomConfig.java CustomService.java

```java
package com.myTraining.core.models;

import com.myTraining.core.models.BookModel;
import com.myTraining.core.services.CustomConfig;
import com.myTraining.core.services.CustomServiceInt;
import org.apache.sling.api.SlingHttpServletRequest;
import org.apache.sling.api.resource.Resource;
import org.apache.sling.models.annotations.Model;
import org.apache.sling.models.annotations.injectorspecific.OSGiService;
import org.apache.sling.models.annotations.injectorspecific.ValueMapValue;
import org.osgi.service.component.annotations.Reference;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import javax.annotation.PostConstruct;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

@Model(adaptables =SlingHttpServletRequest.class , adapters = BookModel.class)
public class BookModelImpl implements BookModel {


    private static final Logger LOG = LoggerFactory.getLogger(BookModelImpl.class);

    @ValueMapValue
    private String loadMoreValue;

    @ValueMapValue
    private String loadMoreCtaName;

    @ValueMapValue
    private List<String> books;
```

BookModel.java BookModelImpl.java CustomConfig.java CustomService.java

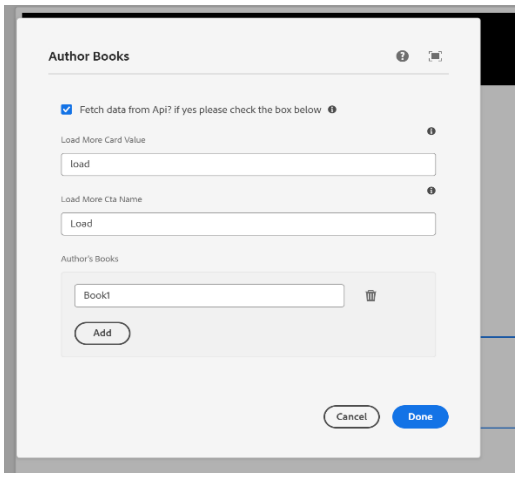
```java
public class BookModelImpl implements BookModel {
    @ValueMapValue
    private List<String> books;

    private  String fetchApi;

    // @Reference
    //CustomServiceInt CSI;

     @OSGiService
     CustomServiceInt CSI;

    public List<String> getBooks() {
        if(books!=null){
            return new ArrayList<String>(books);
        }else{
            return Collections.emptyList();
        }
    }

    @PostConstruct
    protected void init() {
        LOG.info("------------inside Author model init method----------------");
         fetchApi=CSI.getAuthorApi();
        LOG.info("------------fetchApi----------------");

    }


  @Override
    public String fetchApi() {
        return fetchApi;
    }
```

BookModel.java BookModelImpl.java CustomConfig.java CustomService.java

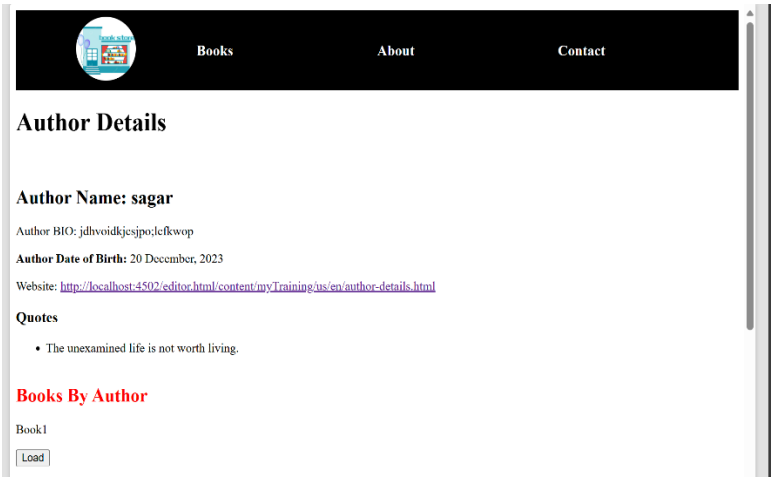
```java
public class BookModelImpl implements BookModel {
    public List<String> getBooks() {
        }else{
            return Collections.emptyList();
        }
    }

    @PostConstruct
    protected void init() {
        LOG.info("------------inside Author model init method----------------");
         fetchApi=CSI.getAuthorApi();
        LOG.info("------------fetchApi----------------");

    }


    @Override
    public String fetchApi() {
        return fetchApi;
    }

    public String getLoadMoreValue() {
        return loadMoreValue;
    }

    public String getLoadMoreCtaName() {
        return loadMoreCtaName;
    }
}
```

Now open AEM and add Book component in the author page

Fill the following fields



Output :


Books
About
Contact
Author Details
Author Name: sagar
Author BIO: jdhvoidkjcsjpo;lcfkwop
Author Date of Birth: 20 December, 2023
Website: http://localhost:4502/editor.html/content/myTraining/us/en/author-details.html
Quotes
The unexamined life is not worth living.
Books By Author
Book1
Load