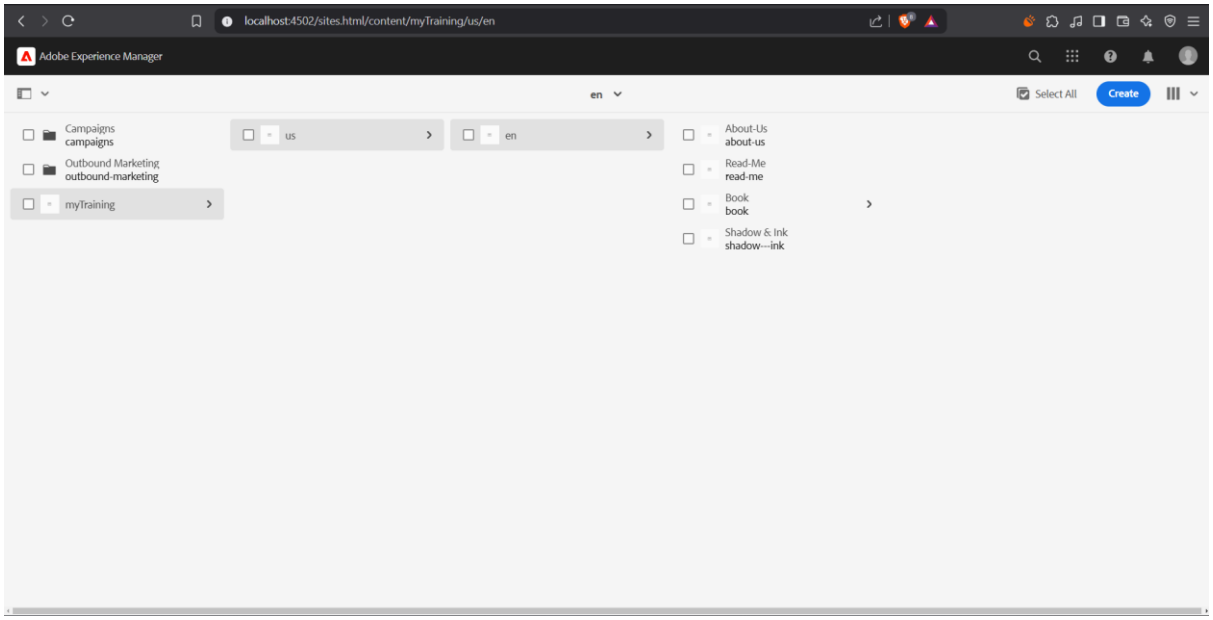
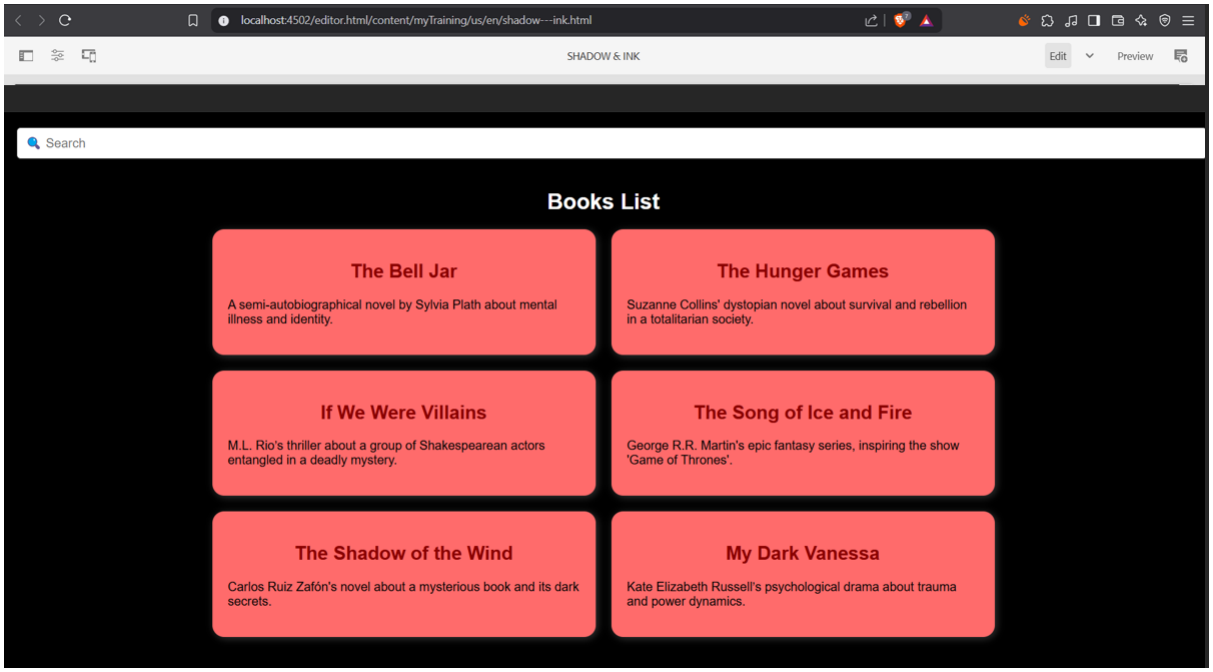
CARD DISPLAY OF BOOK :





BookModel.java

```
package com.example.core.models;


import com.adobe.cq.sightly.WCMUsePojo;
```

```
import com.adobe.granite.asset.api.Asset;
import com.adobe.granite.query.QueryBuilder;
import com.adobe.granite.query.Query;
import com.adobe.granite.query.QueryResult;
import org.apache.sling.api.resource.Resource;
import org.apache.sling.models.annotations.Model;
import org.apache.sling.models.annotations.DefaultInjectionStrategy;
import org.apache.sling.models.annotations.injectorspecific.ValueMapValue;
import org.apache.sling.models.annotations.injectorspecific.SlingObject;
import org.apache.sling.models.annotations.injectorspecific.OSGiService;

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

@Model(adaptables = Resource.class, defaultInjectionStrategy = DefaultInjectionStrategy.OPTIONAL)
public class BookModel {

    @ValueMapValue
    private String bookRootPath;

    @ValueMapValue
    private String loadMoreLabel;

    @SlingObject
    private Resource resource;

    @OSGiService
    private QueryBuilder queryBuilder;
```

```
    private List<Book> books;

    @PostConstruct
    protected void init() {
        books = fetchBooks();
    }

    private List<Book> fetchBooks() {
        List<Book> bookList = new ArrayList<>();
        if (bookRootPath != null) {
            String queryString = "SELECT * FROM [cq:Page] AS s WHERE ISDESCENDANTNODE(s, '" + bookRootPath + "') ORDER BY [jcr:content/jcr:lastModified] DESC";
            Query query = queryBuilder.createQuery(queryString, resource.getResourceResolver().adaptTo(Session.class));
            QueryResult result = query.getResult();
            result.getNodes().forEachRemaining(node -> {
                String title = node.getProperty("jcr:title").getString();
                String imagePath = node.getProperty("imagePath").getString();
                String description = node.getProperty("description").getString();
                String publishDate = node.getProperty("publishDate").getString();
                bookList.add(new Book(title, imagePath, description, publishDate));
            });
        }
        return bookList;
    }

    public List<Book> getBooks() {
        return books;
    }
```

```
public String getLoadMoreLabel() {
    return loadMoreLabel;
}

public static class Book {
    private String title;
    private String image;
    private String description;
    private String publishDate;

    public Book(String title, String image, String description, String publishDate) {
        this.title = title;
        this.image = image;
        this.description = description;
        this.publishDate = publishDate;
    }

    public String getTitle() {
        return title;
    }

    public String getImage() {
        return image;
    }

    public String getDescription() {
        return description;
    }
```

```
        public String getPublishDate() {
            return publishDate;
        }
    }
}
```

Book.html

```
<sly data-sly-use.model="com.example.core.models.BookModel"/>

<div class="book-container">
    <div class="book-list">
        <sly data-sly-list.book="${model.books}">
            <div class="book-card">
                <img src="${book.image}" alt="${book.title}" />
                <h3>${book.title}</h3>
                <p>${book.description}</p>
                <p class="date">Published on: ${book.publishDate}</p>
            </div>
        </sly>
    </div>
    <button class="load-more">${model.loadMoreLabel}</button>
</div>
```

Book.css

```
.book-container {
    display: flex;
    flex-direction: column;
    align-items: center;
}
```

```
.book-list {
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
}

.book-card {
    width: 250px;
    padding: 10px;
    margin: 10px;
    border: 1px solid #ccc;
    text-align: center;
}

.book-card img {
    width: 100%;
    height: auto;
}

.book-card .date {
    font-size: 12px;
    color: gray;
}

/* Horizontal Layout */
.horizontal .book-list {
    flex-direction: row;
}

.vertical .book-list {
```

```
    flex-direction: column;
}
```

BookEventHandler.java

```
package com.example.core.listeners;

import org.apache.sling.api.SlingConstants;
import org.osgi.service.component.annotations.Component;
import org.osgi.service.event.Event;
import org.osgi.service.event.EventHandler;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

@Component(
        service = EventHandler.class,
        property = {
                "event.topics=" + SlingConstants.TOPIC_RESOURCE_ADDED
        }
)
public class BookEventHandler implements EventHandler {

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

    @Override
    public void handleEvent(Event event) {
        String path = (String) event.getProperty(SlingConstants.PROPERTY_PATH);
        if (path.startsWith("/content/books")) {
            LOG.info("New book page created: {}", path);
        }
    }
```

}