**DATE : 18-03-2025**

**Maven and AEM Project Structure Guide**

**1. Maven Life Cycle**

**Maven follows a structured build life cycle consisting of phases:**

* **Clean: Removes target directory and compiled files.**
* **Validate: Checks the correctness of the project.**
* **Compile: Compiles the source code.**
* **Test: Runs unit tests.**
* **Package: Bundles the project into a JAR/WAR.**
* **Verify: Ensures the integrity of the package.**
* **Install: Installs the package in the local repository.**
* **Deploy: Deploys the package to a remote repository.**

**Command to Run Maven Life Cycle:**

**mvn clean install**

**2. What is pom.xml File and Why Do We Use It?**

**POM (Project Object Model) XML is the core configuration file in a Maven project. It defines:**

* **Project Information**
* **Dependencies (libraries required for the project)**
* **Plugins (to perform tasks like compilation, packaging, deployment)**
* **Build Configuration**

**Example pom.xml:**

**<project xmlns="http://maven.apache.org/POM/4.0.0" ...>**

**<modelVersion>4.0.0</modelVersion>**

**<groupId>com.myTraining</groupId>**

**<artifactId>aem-project</artifactId>**

**<version>1.0-SNAPSHOT</version>**

**<packaging>pom</packaging>**

**</project>**

**3. How Dependencies Work?**

**Dependencies define external libraries required for a project. Maven fetches these from a central repository.**

**Example Dependency in pom.xml:**

**<dependencies>**

**<dependency>**

**<groupId>org.apache.sling</groupId>**

**<artifactId>org.apache.sling.api</artifactId>**

**<version>2.20.0</version>**

**</dependency>**

**</dependencies>**

**4. Check the Maven Repository**

**Maven repositories are used to store dependencies.**

* **Local Repository (~/.m2/repository/)**
* **Central Repository (Maven’s default repository)**
* **Remote Repository (hosted by organizations)**

**Command to Check Local Repo:**

**ls ~/.m2/repository/**

**5. How All Modules Build Using Maven?**

**AEM projects use a multi-module structure, and all modules can be built together using:**

**mvn clean install**

**This will compile and package all modules in the project.**

**6. Can We Build a Specific Module?**

**Yes, use the -pl (project list) flag to build a specific module.**

**Example: Build only core module:**

**mvn clean install -pl core -am**

**7. Role of ui.apps, ui.content, and ui.frontend Folder**

* **ui.apps: Stores AEM components, templates, and Java code.**
* **ui.content: Contains sample content like pages and assets.**
* **ui.frontend: Holds frontend assets (CSS, JS, client-side libraries).**

**8. Why Are We Using Run Mode?**

**Run modes allow different configurations for different environments (dev, stage, prod).**

**Example Command to Run AEM in Specific Mode:**

**java -jar aem-author-p4502.jar -r author,dev**

**9. What is Publish Environment?**

**The publish environment (port 4503) serves content to end-users. Content is replicated from the author environment (4502) to the publish instance.**

**10. Why Are We Using Dispatcher?**

**The Dispatcher is used for:**

* **Caching: Improves performance.**
* **Security: Filters requests before reaching AEM.**
* **Load Balancing: Distributes traffic across multiple instances.**

**11. From Where Can We Access crx/de?**

**The CRX/DE (Content Repository Extreme Developer Edition) is accessed via:**

**http://localhost:4502/crx/de**

**This allows developers to view and modify AEM content structures.**

**DATE : 19-03-2025**

**Digital Asset Management (DAM) and Tasks Execution Guide**

**1. What is DAM and Why Do We Use It?**

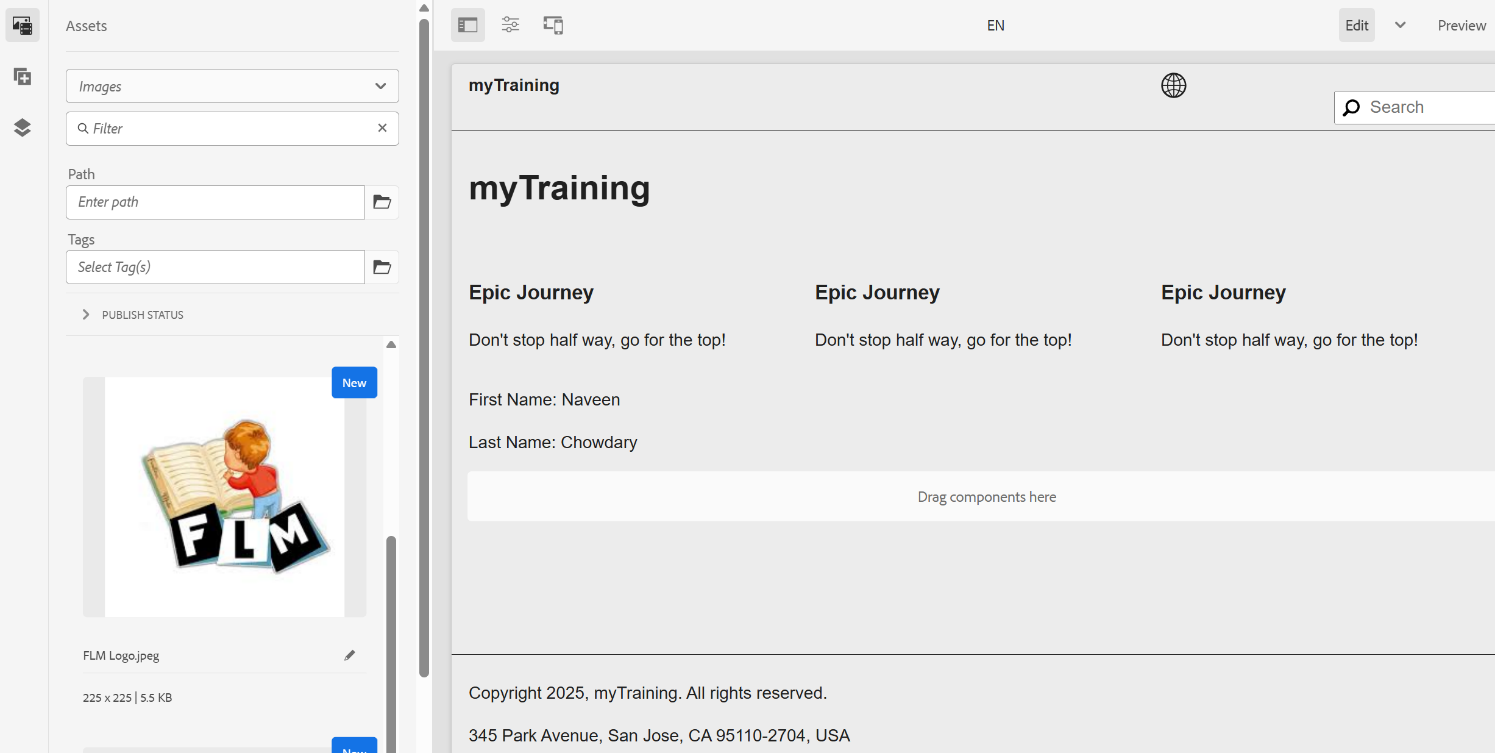
**Digital Asset Management (DAM) is a system that allows organizations to store, manage, and distribute digital assets (such as images, videos, documents) efficiently. In Adobe Experience Manager (AEM), DAM is used to:**

* **Organize and centralize digital assets.**
* **Enable easy retrieval and reuse of assets.**
* **Optimize asset delivery with renditions.**
* **Ensure version control and access permissions.**

**2. Create a Folder and Upload Images in DAM**

**Steps:**

1. **Navigate to AEM > Assets > Files.**
2. **Create a new folder at the path: /content/dam/myTraining/us/en-us.**
3. **Upload two images inside this folder.**
4. **Open a page in AEM Sites.**
5. **Drag and drop the Image Component onto the page.**
6. **Select the uploaded images from DAM and author them into the component.**

****

**3. What Are Renditions?**

**Renditions are different versions of an image automatically created by AEM DAM. These variations help optimize asset delivery for different devices and use cases.**

**Steps to Check Renditions:**

1. **Navigate to AEM > Assets > Files.**
2. **Open one of the uploaded images.**
3. **Click on Renditions to view the different sizes available.**

**4. Add FirstName and LastName Fields in HelloWorld Component**

**Steps:**

1. **Open the HelloWorld component dialog in AEM Component Editor.**
2. **Add two fields:**
   * **First Name (firstName)**
   * **Last Name (lastName)**
3. **Modify helloworld.html to print these values using properties.**

**Example Code (helloworld.html):**

**<p>First Name: ${properties.firstName}</p>**

**<p>Last Name: ${properties.lastName}</p>**

**5. Use @ValueMapValue in HelloWorldModel**

**Modify the Sling model (HelloWorldModel.java) to use @ValueMapValue.**

**Example Code:**

**@Model(adaptables = Resource.class)**

**public class HelloWorldModel {**

**@ValueMapValue**

**private String firstName;**

**@ValueMapValue**

**private String lastName;**

**public String getFirstName() {**

**return firstName;**

**}**

**public String getLastName() {**

**return lastName;**

**}**

**}**

**6. Why Use Package Manager? (Creating Packages)**

**Package Manager is used to:**

* **Export/import components and assets between AEM instances.**
* **Ensure consistent deployment.**
* **Backup assets and configurations.**

**Create Packages:**

1. **DAM Package (Images)**
   * **Go to AEM > Tools > Deployment > Package Manager.**
   * **Click Create Package.**
   * **Name it DAM-Images-Package.**
   * **Add the path: /content/dam/myTraining/us/en-us.**
   * **Build and download the package.**
2. **HelloWorld Component Package**
   * **Create another package named HelloWorld-Component-Package.**
   * **Add the path: /apps/myTraining/components/helloworld.**
   * **Build and download the package.**

**7. Configure Replication Agent and Publish Page**

**Steps:**

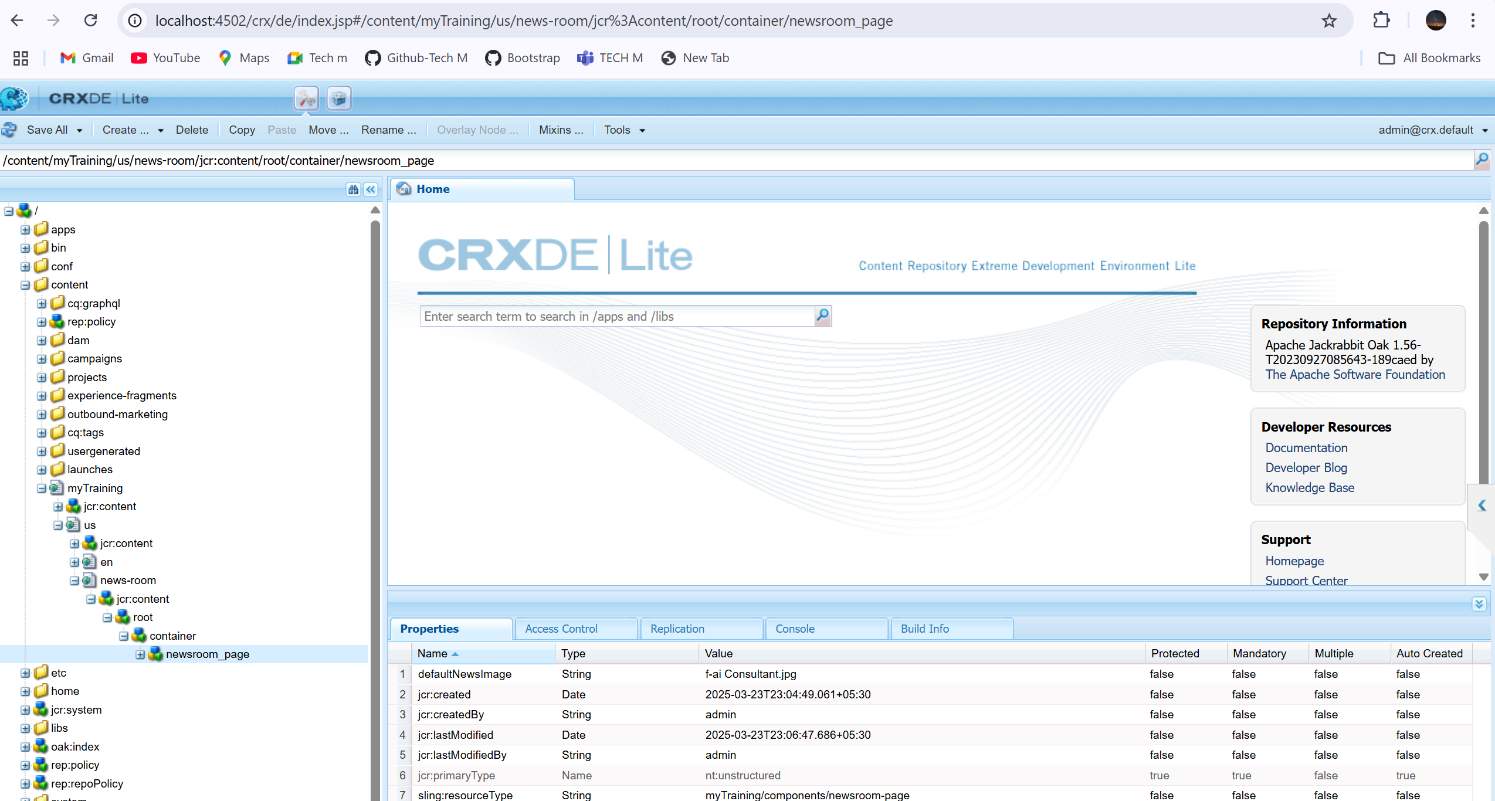
1. **Navigate to Replication Agents:**
   * **Go to AEM > Tools > Deployment > Replication.**
   * **Select Agents on Author.**
   * **Edit the default replication agent.**
   * **Set Transport URI to http://localhost:4503/bin/receive (assuming default setup).**
   * **Click Test Connection and Save.**
2. **Publish a Page:**
   * **Navigate to AEM > Sites.**
   * **Select the page and click Publish.**

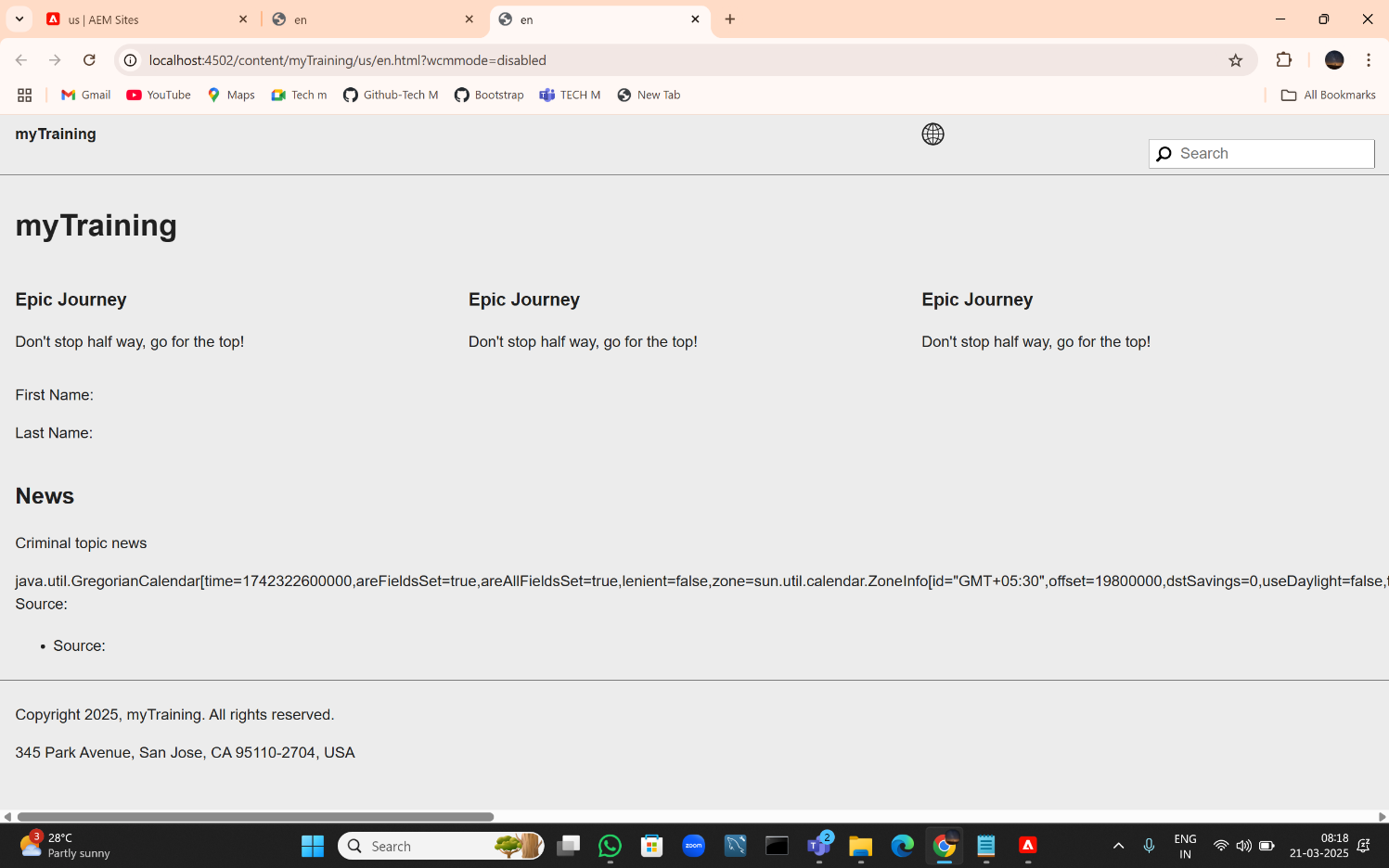
**Command for Replication Test:**

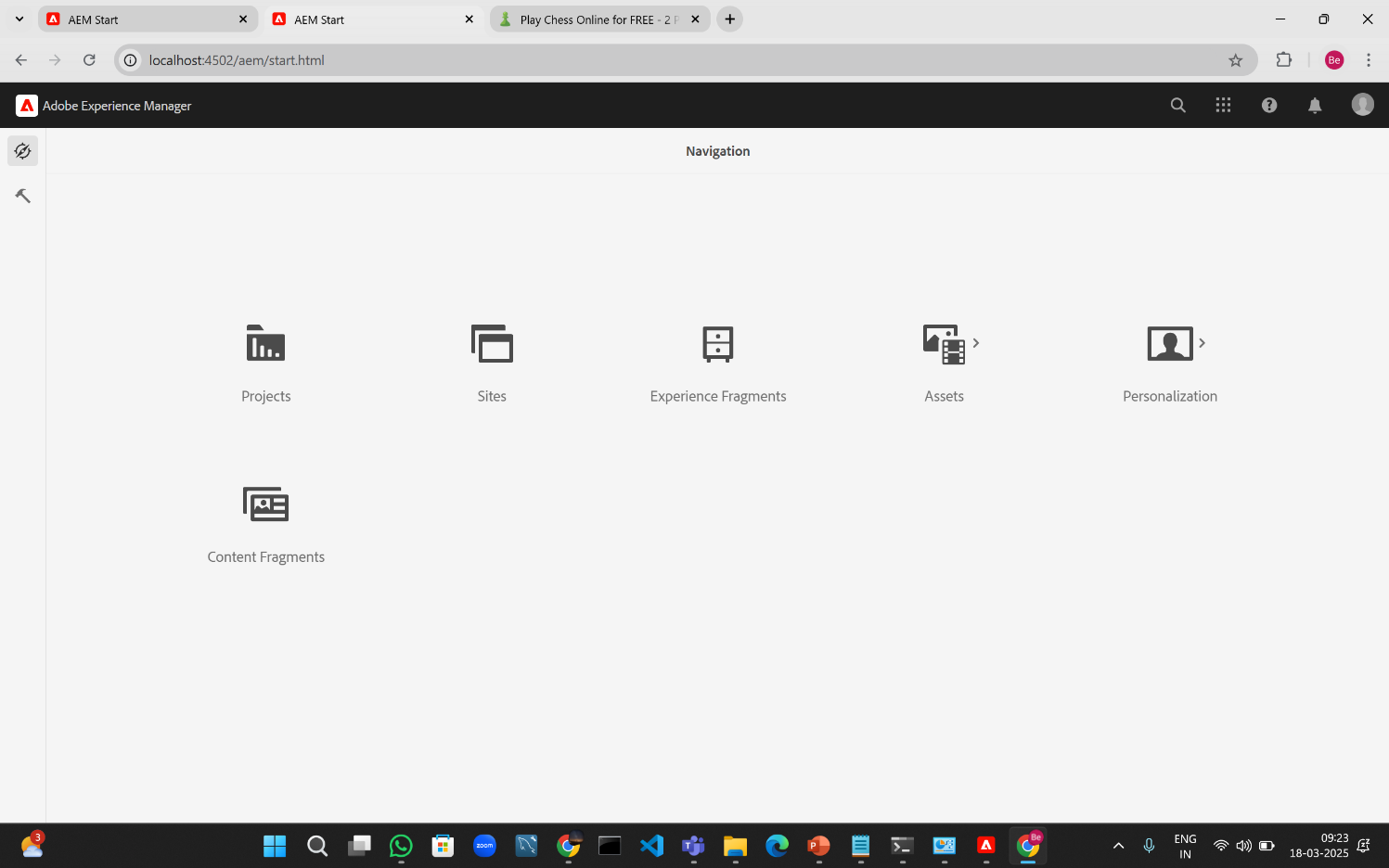
**curl -u admin:admin -X POST http://localhost:4502/bin/replicate.json -d "path=/content/my-page&type=activate"**

**Final Note:**

* **If issues arise, check logs in AEM Error.log (/crx-quickstart/logs)**
* **For further debugging, use AEM Web Console (http://localhost:4502/system/console).**

****

****

****

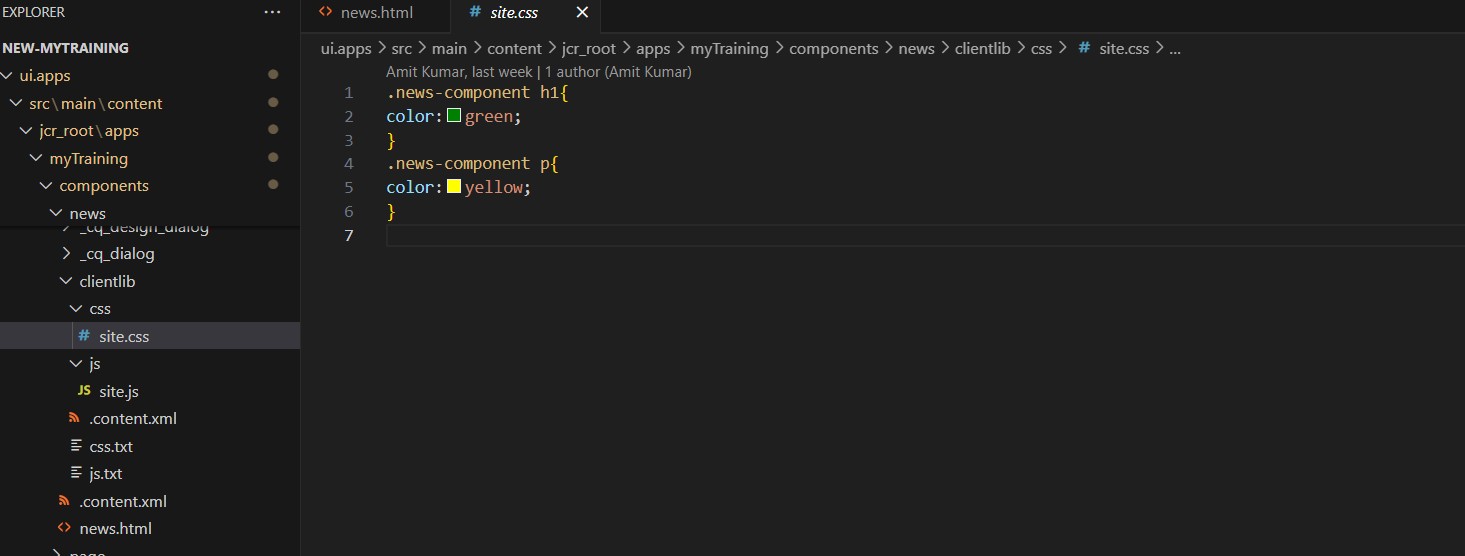
**DATE : 20-03-2025**

4.Apply Styling (CSS)

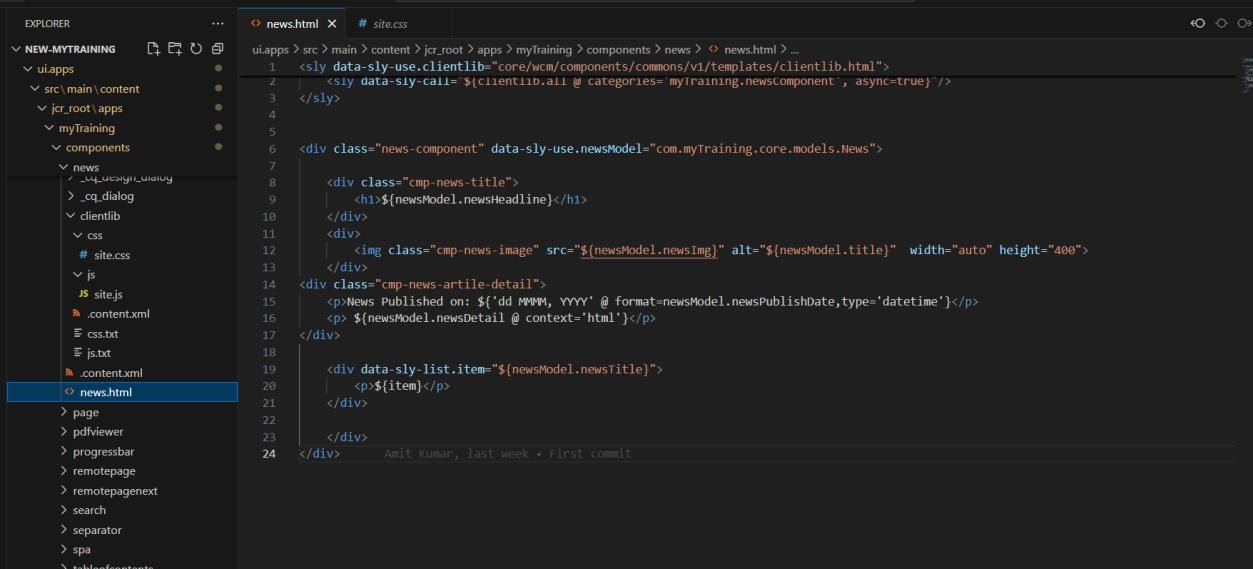
Inspect the rendered page using **Developer Tools (F12) > Elements Tab** and check:

* **Heading (h2) is Green (color: green)**.
* **News Detail (<p>) is Yellow (color: yellow)**.
* **Date is Black (color: black)**.

Css.txt

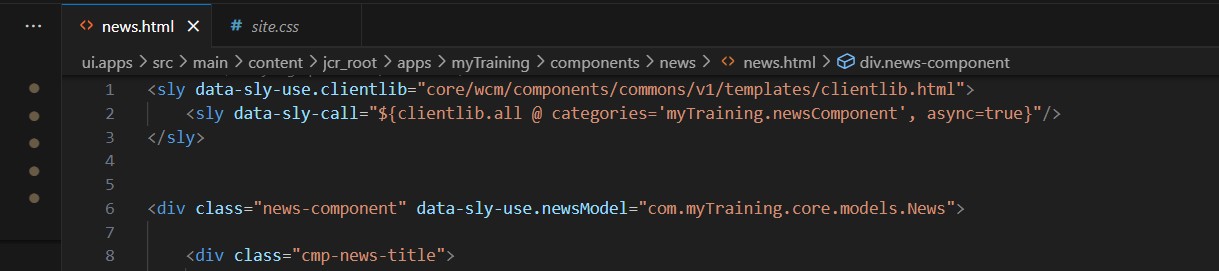


News.html



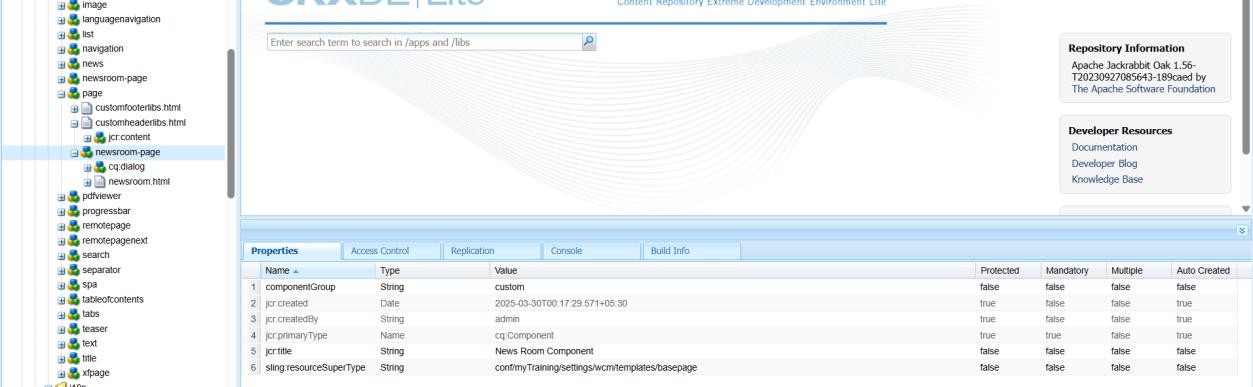
**8. Extra ClientLibs (extraClientLibs) How to Check:**

• Open the multi-field component’s **HTL (HTML Template Language) file**.

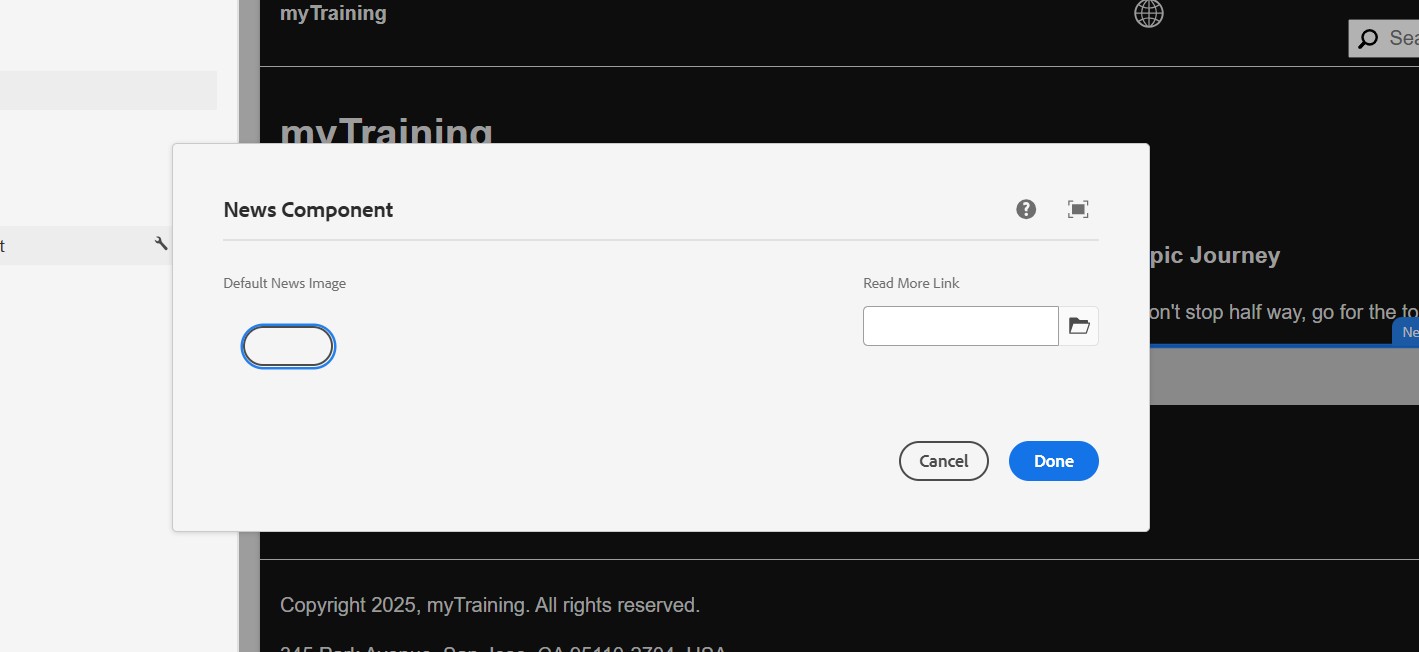


**DATE: 21-03-2025**

1. Create a News Room Page Component using the Base Page Component



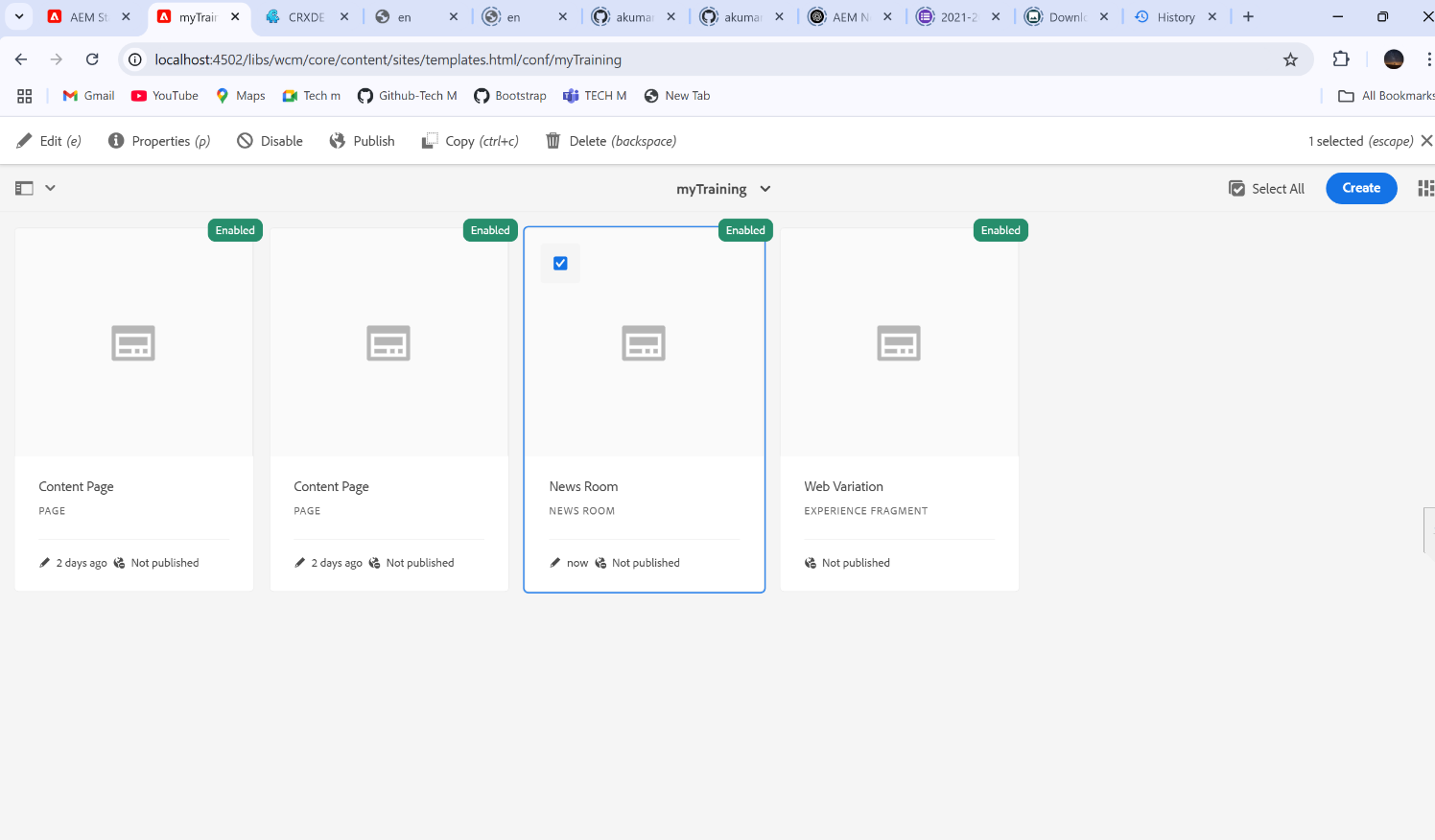
1. Create Custom Page Property: "NEWS Configurations"
2. Add "Read More" CTA for News Articles



1. **Create News Room Template Type using News Room Page Component**

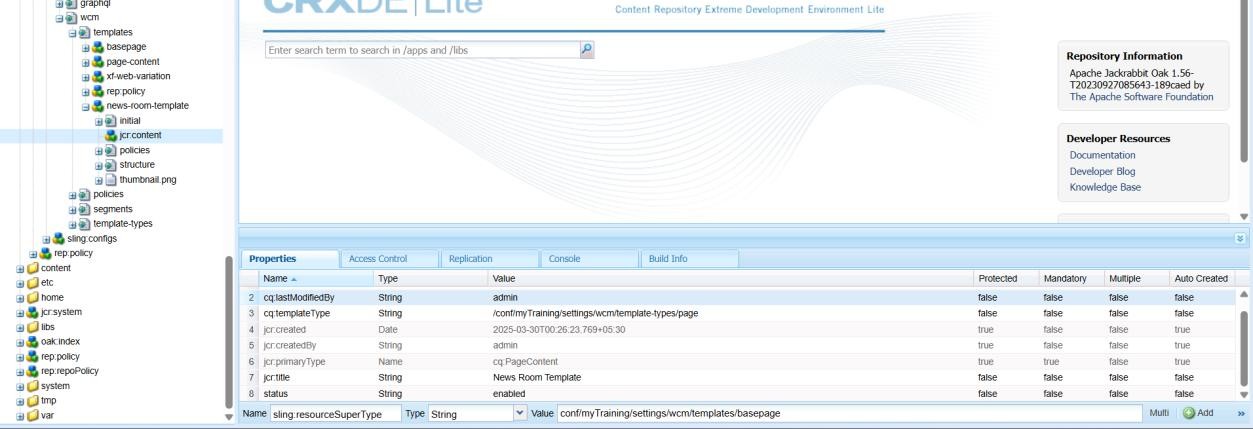
**Steps to Implement:**

* + Go to **AEM > Tools > General > Template Types**.
  + Create a new Template Type named **News Room Template Type**.
  + Set the **Allowed Resource Type** as your-project/components/page/newsroompage.



**How to Verify:**

* + Go to **Template Editor** and check if the template type appears.
  + Try creating a template using this type.



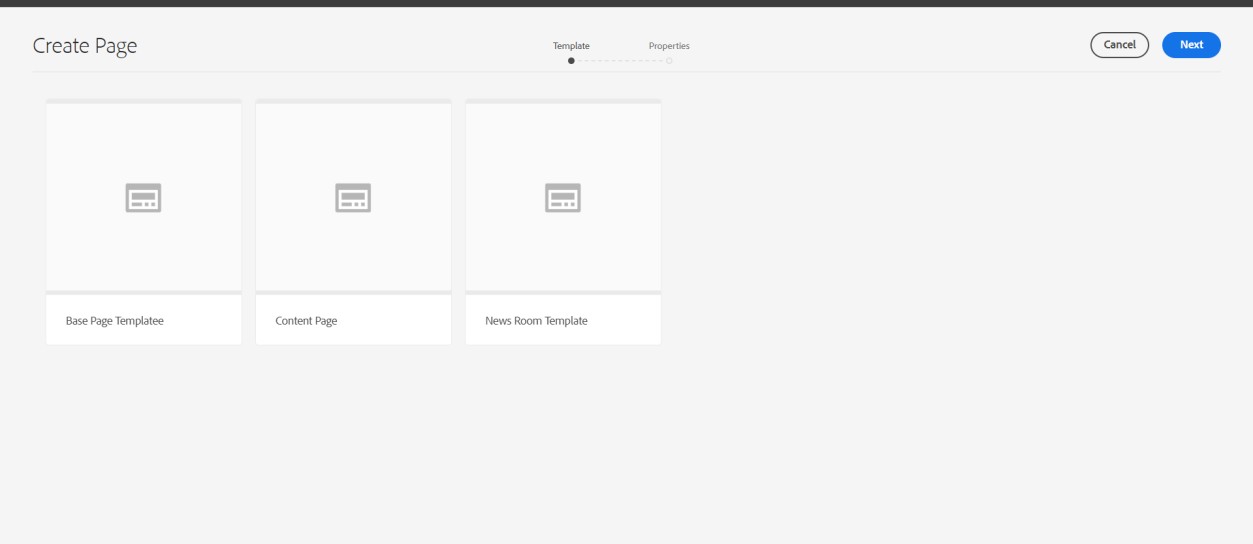
1. **Create News Room Template using News Room Template Type**

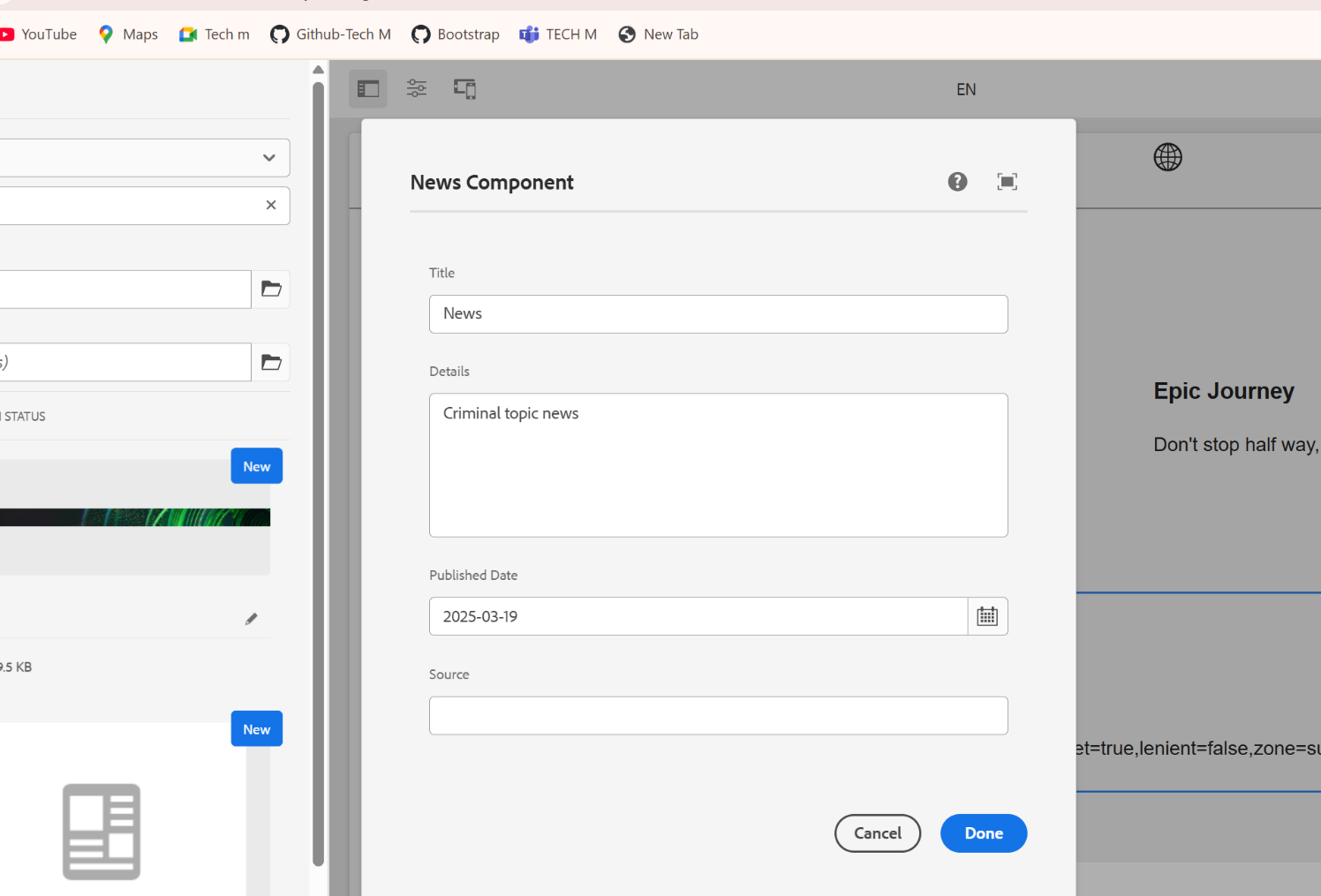
**Steps to Implement:**

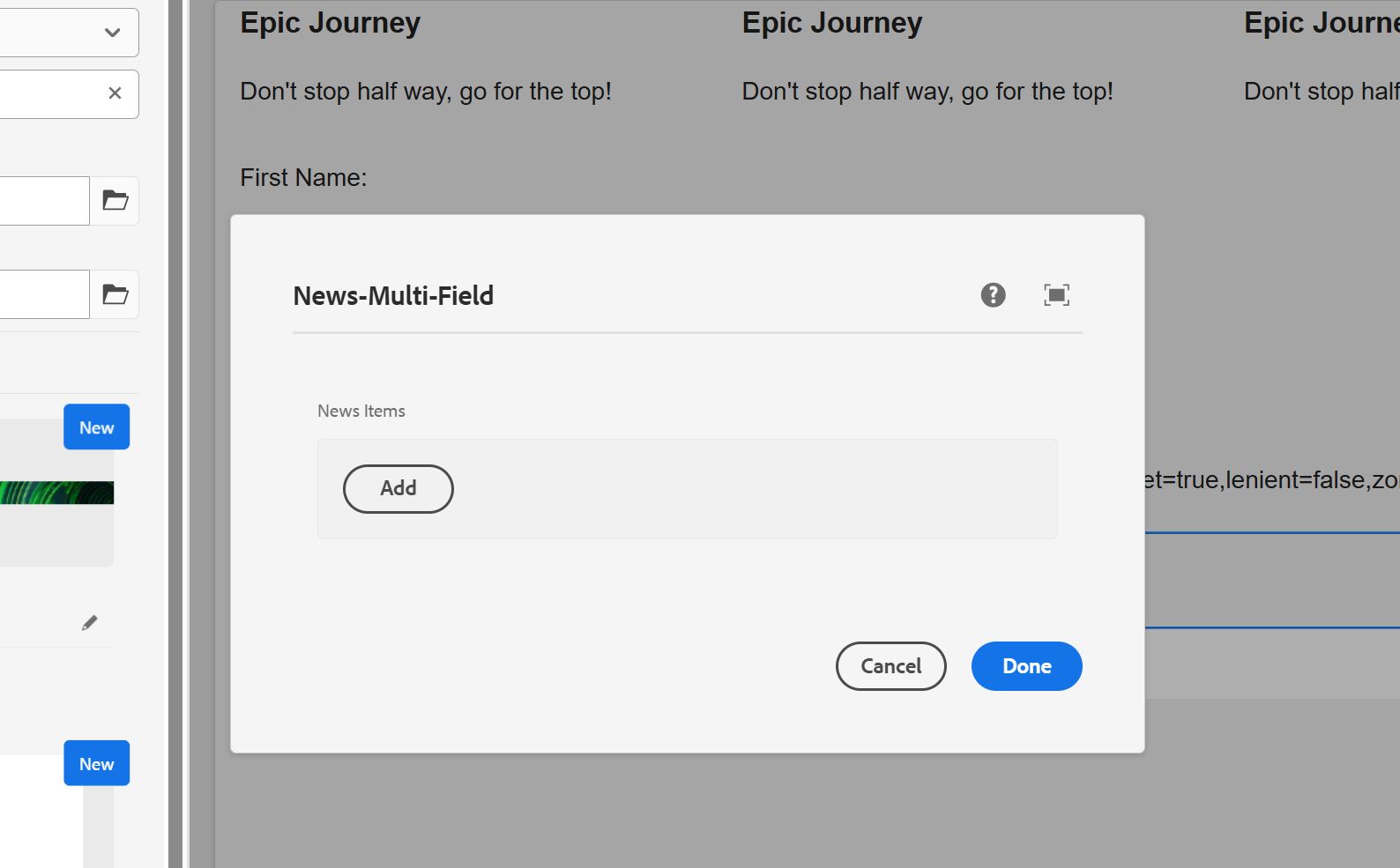
* + In **AEM > Templates**, create a new template using **News Room Template Type**.
  + Define the structure (Header, News Listing, Footer, etc.).

**How to Verify:**

* + Check if the template appears under **Templates** in AEM.
  + Create a page using this template and verify the structure.



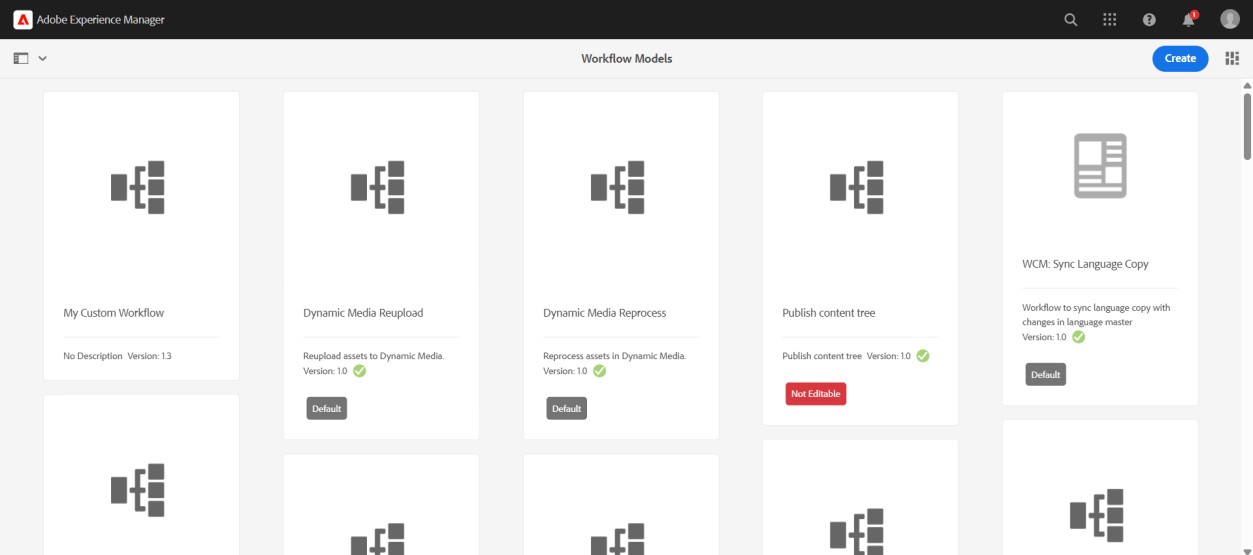


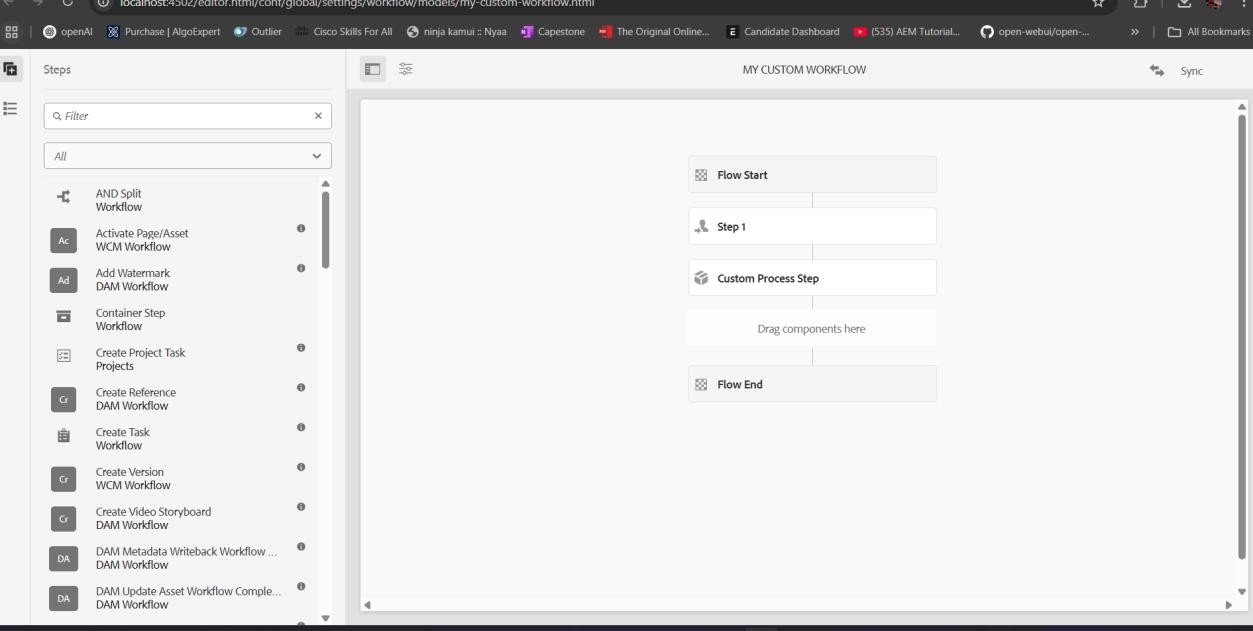
****

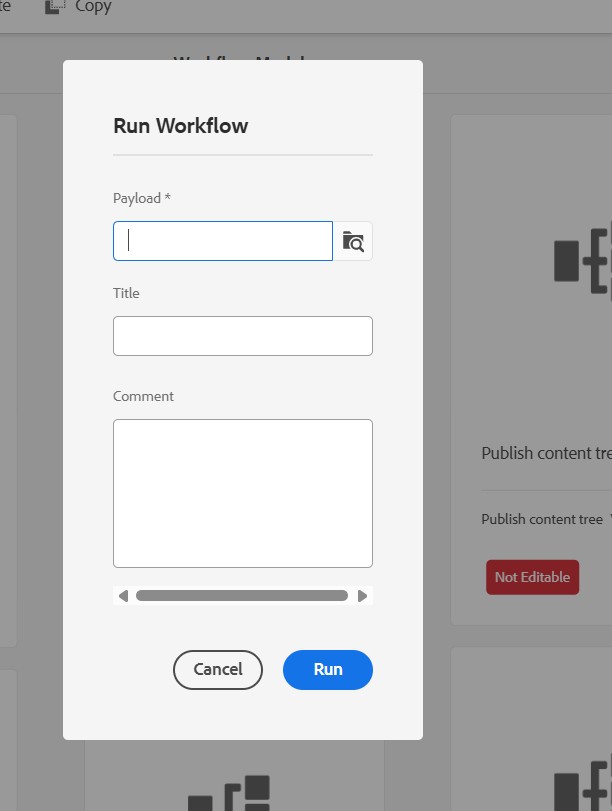
**DATE : 25-03-2025**

**Steps to Check:**

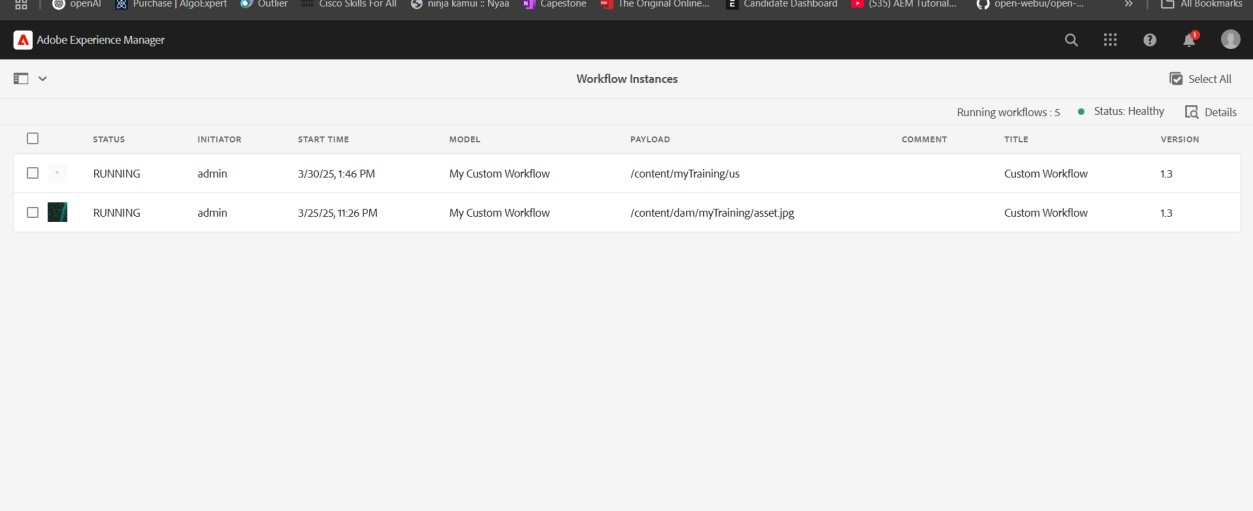
* + Go to **AEM Start → Tools → Workflow → Models**.
  + Check if the **"my custom workflow"** is listed.
  + Open it and ensure the **custom workflow process step** is configured properly.
  + Trigger the workflow on a page by:
    - Navigating to **Sites → Select a Page**.
    - Click on **Page Properties → Start Workflow** and choose your custom workflow.
  + **Check Logs**: o Open AEM **error.log** (crx-quickstart/logs/error.log).
    - Look for log statements that print the **page title**.

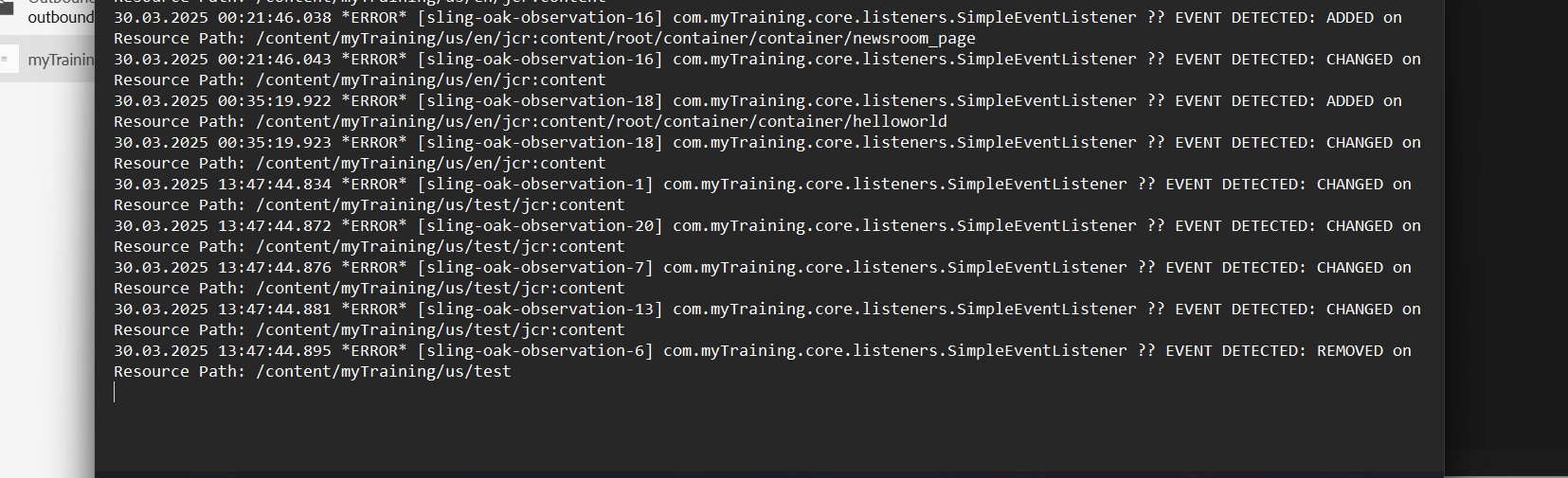




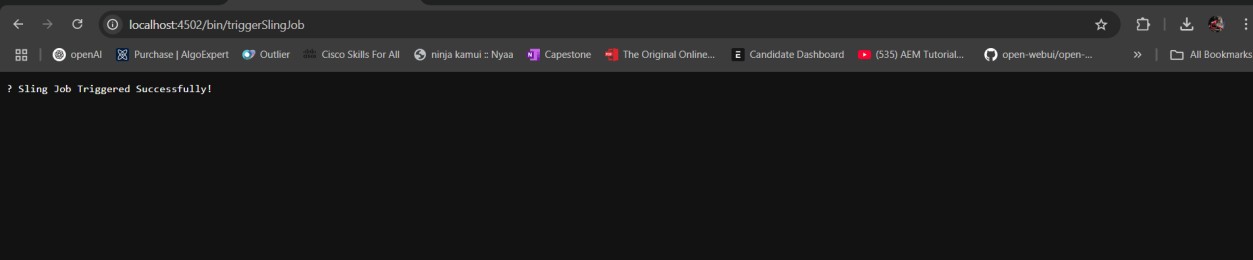


1. **Verify Event Handler Execution Steps to Check:** 
   * Perform an action that triggers the event (e.g., creating, modifying, or deleting a page or an asset).
   * **Check Logs:** o Open error.log and look for log messages related to the resource path.

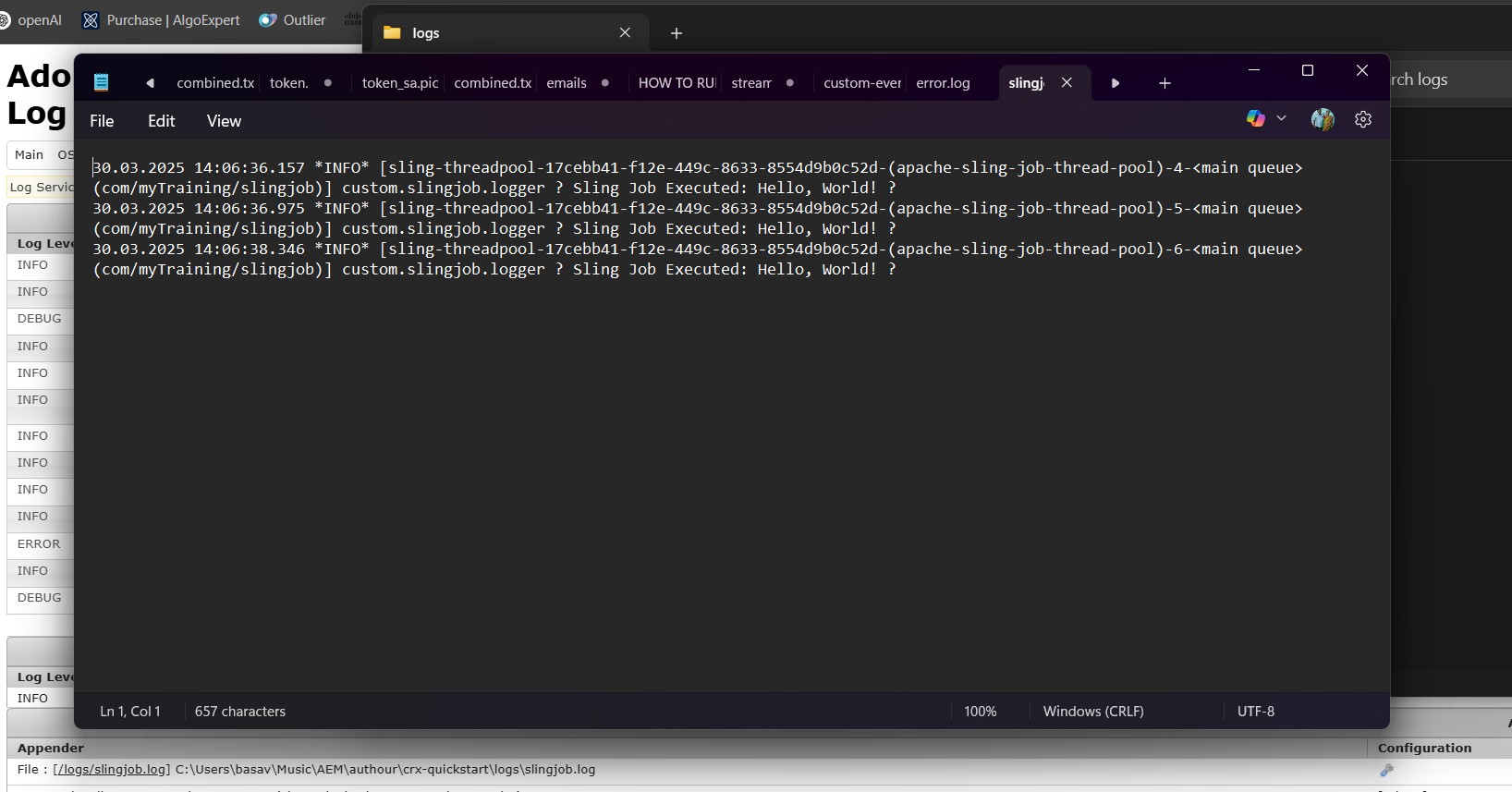




1. **Verify Sling Job Execution Steps to Check:** 
   * Go to **AEM Start → Tools → Operations → Jobs**.
   * Ensure your custom **Sling Job** is listed.
   * **Check Logs:**
     + Open error.log and look for **"Hello World"** message.



Error.log



1. **Verify Scheduler Execution Steps to Check:** 
   * Wait for **5 minutes** after deploying the scheduler.
   * **Check Logs:** o Open error.log and look for **"Yellow World"** printed at every 5-minute interval.

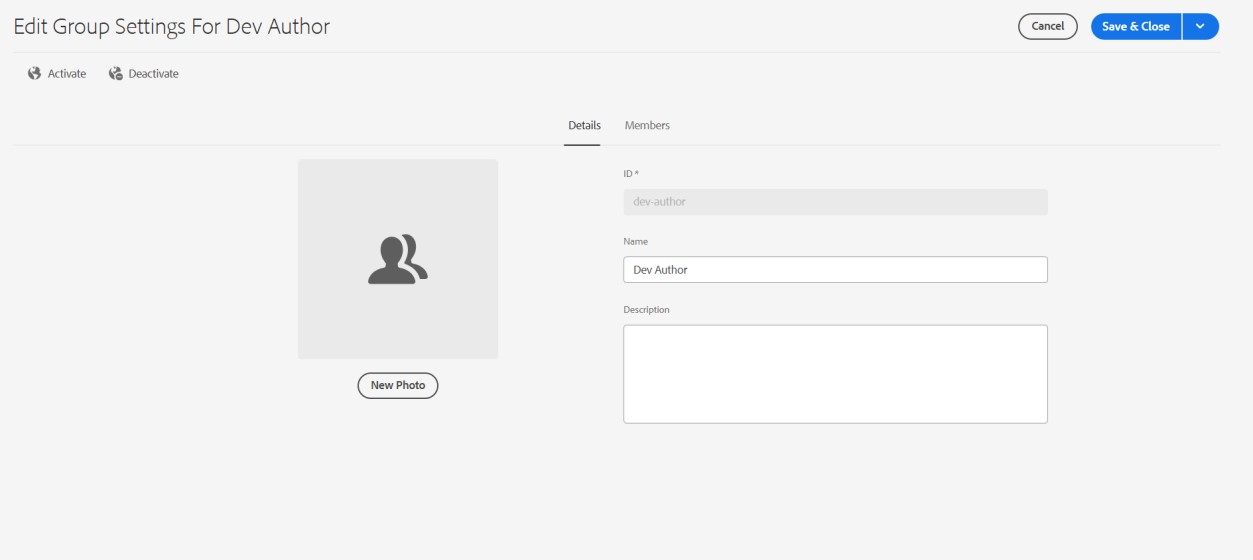
**Expected Output in Logs:**

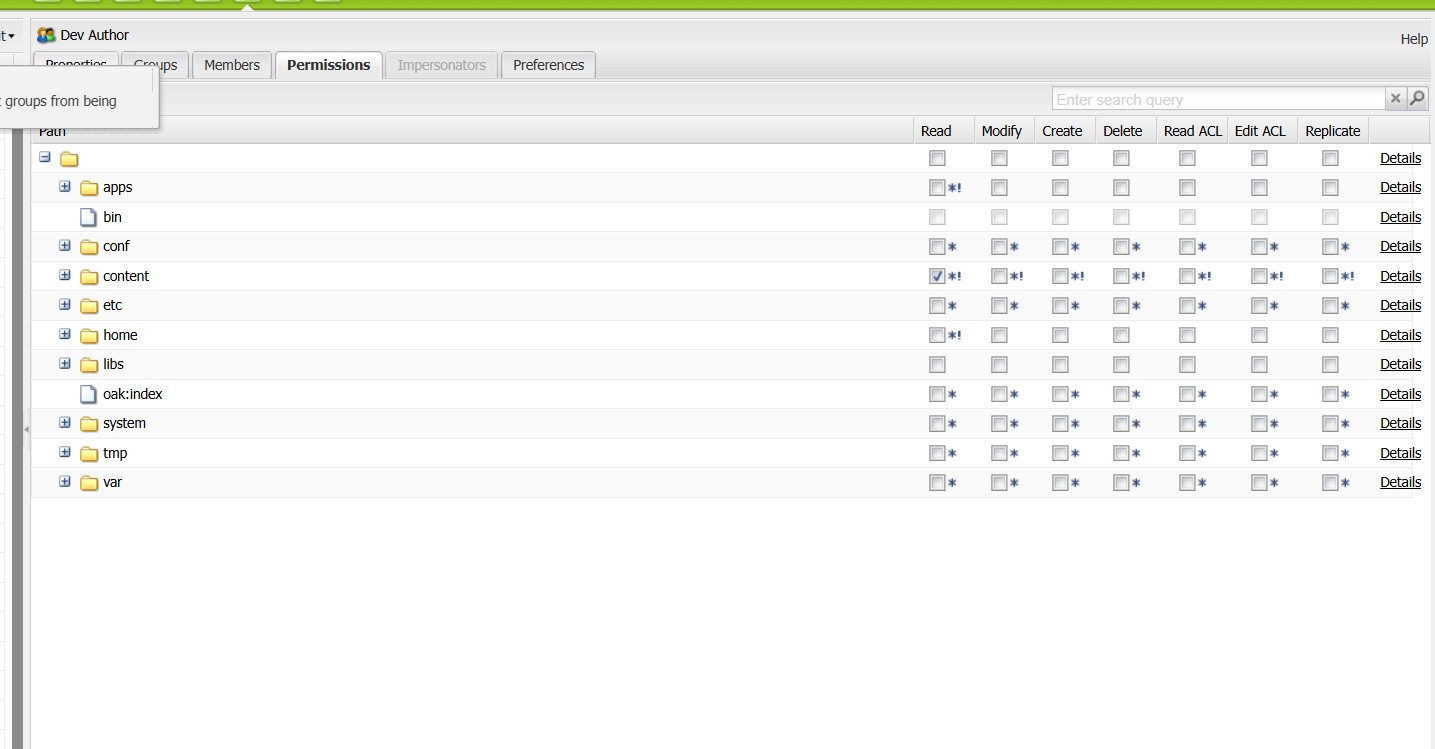
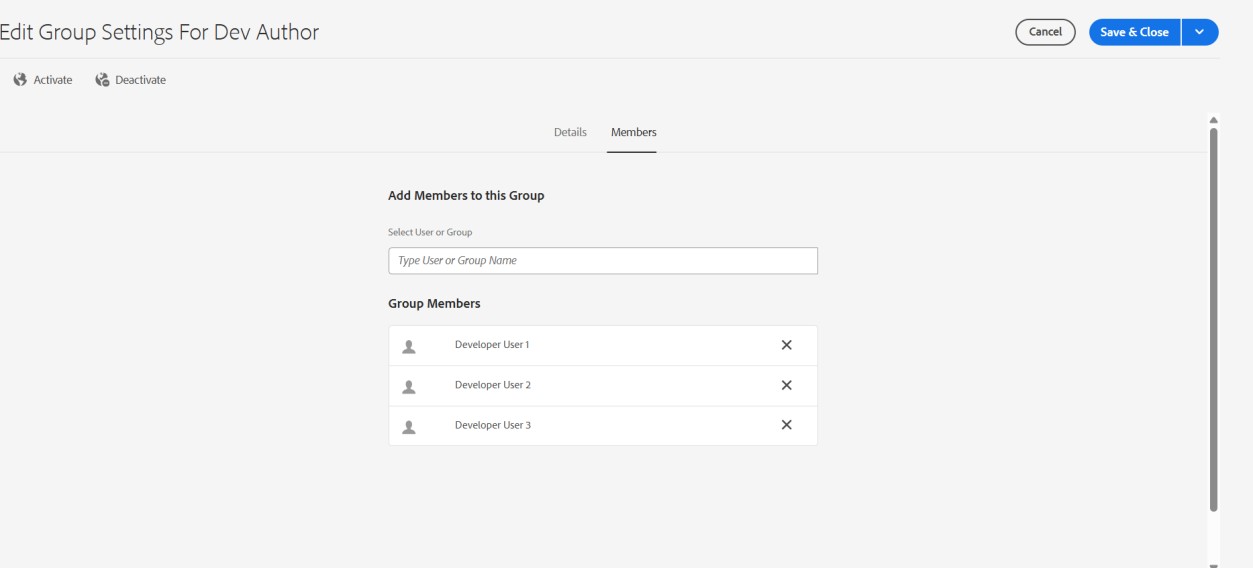
Yellow World printed by Scheduler

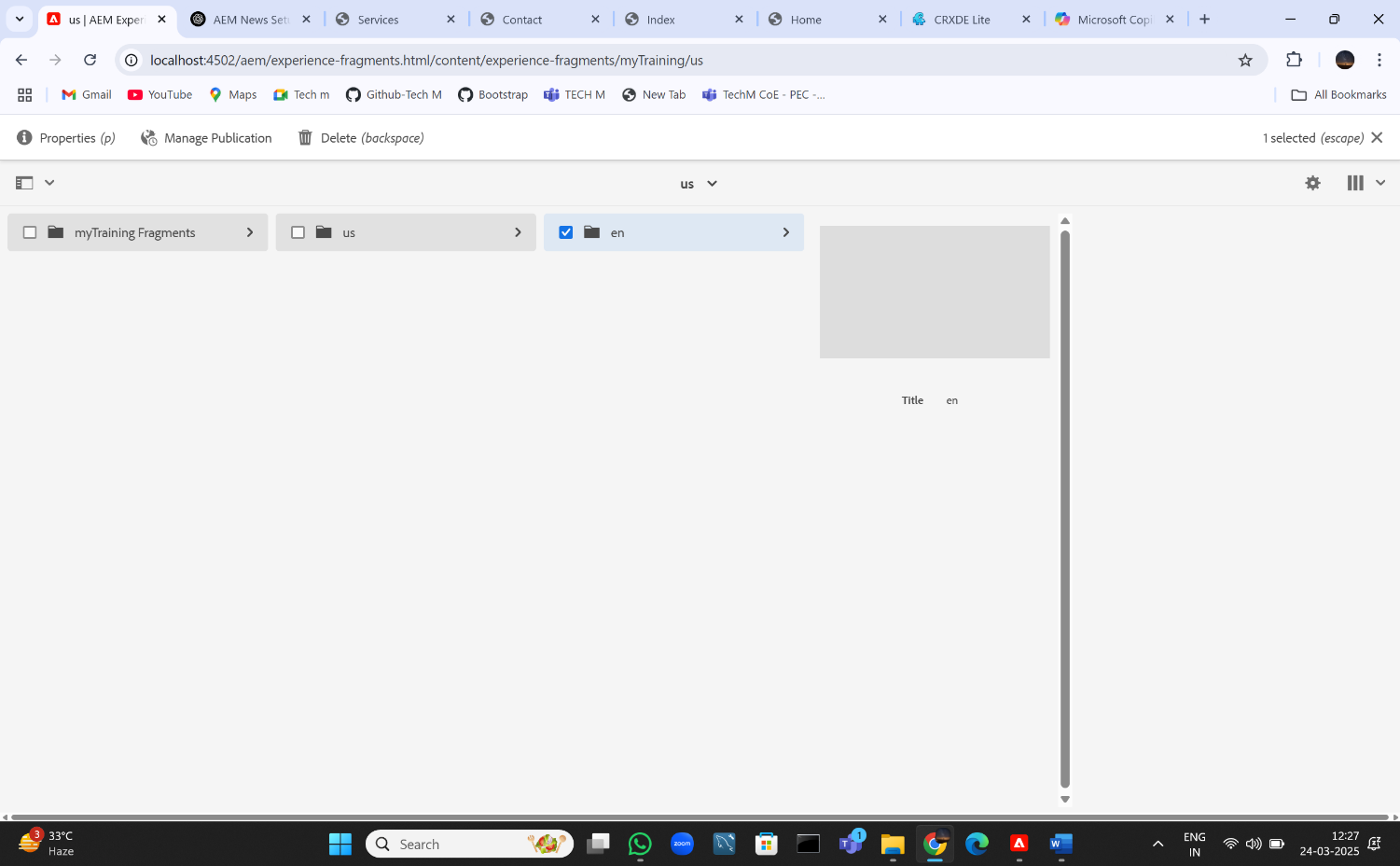
1. **Verify User Group & Permissions Steps to Check:** 
   * Go to **AEM Start → Tools → Security → Groups**.
   * Find **"Dev Author"** group and open it.
   * Ensure the three users are **added**.
   * Check **Permissions:**
     + Go to **/useradmin** (http://localhost:4502/useradmin). o Select the group and verify **Read access** is set for /content and /dam.
     + Ensure **Replication Access** is enabled under **Replicator Permissions**.

**Test Permissions by Logging in as a User in the Group**

* + Login with one of the created users.
  + Try to **view** a page under /content and an asset in /dam.
  + Try **editing** or **deleting** content (should be restricted).
  + Try **replicating** content (should be allowed).



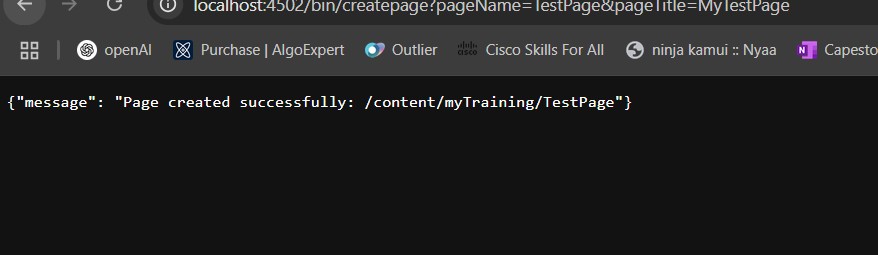


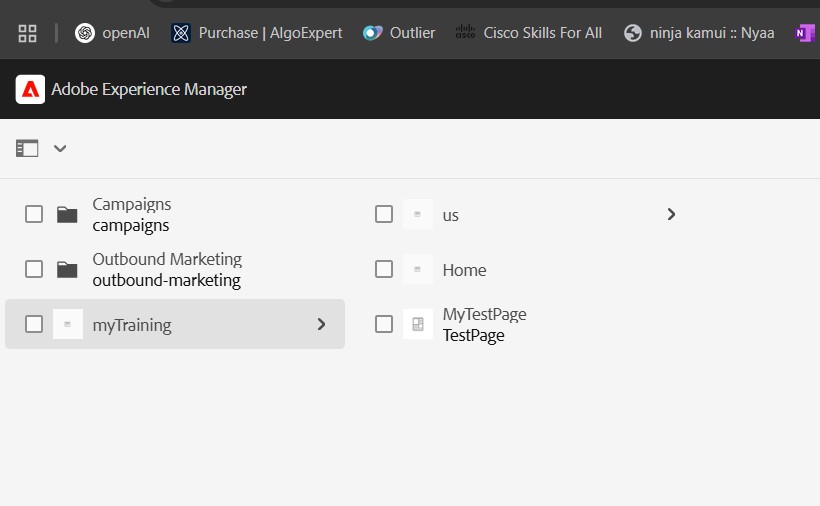
****

**DATE : 26-03-2025**

1. **SampleServlet (resourceType-based registration) Check Output:** 
   * **URL Format:** http://localhost:4502/content/samplepage.sample.json • Replace samplepage with the actual page where you registered the servlet.
   * If it returns the expected JSON response, the servlet is working.

1. **CreatePageServlet (Registered via path) Check Output:**

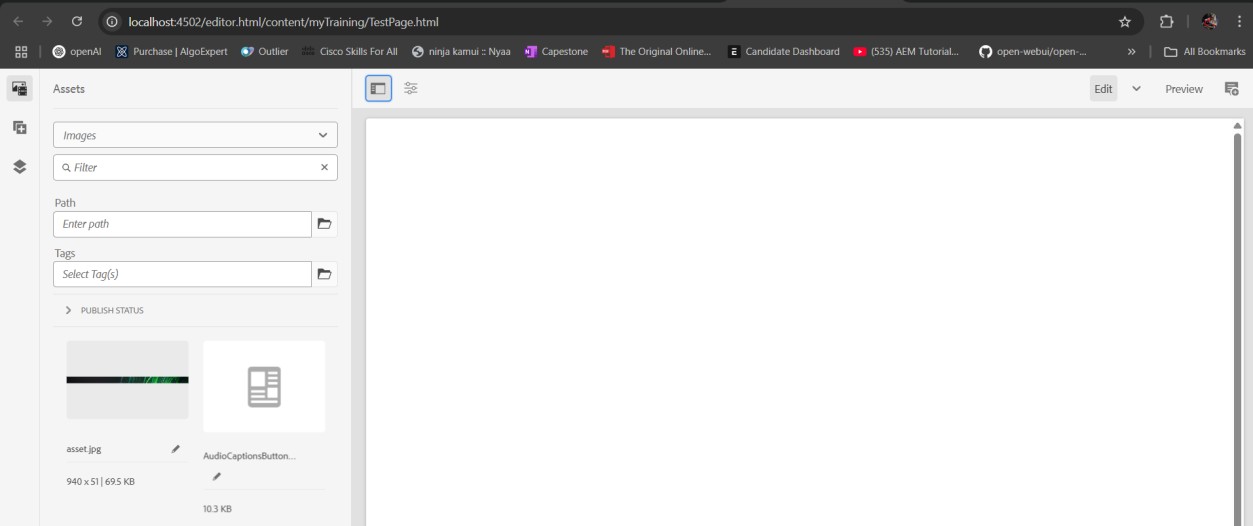




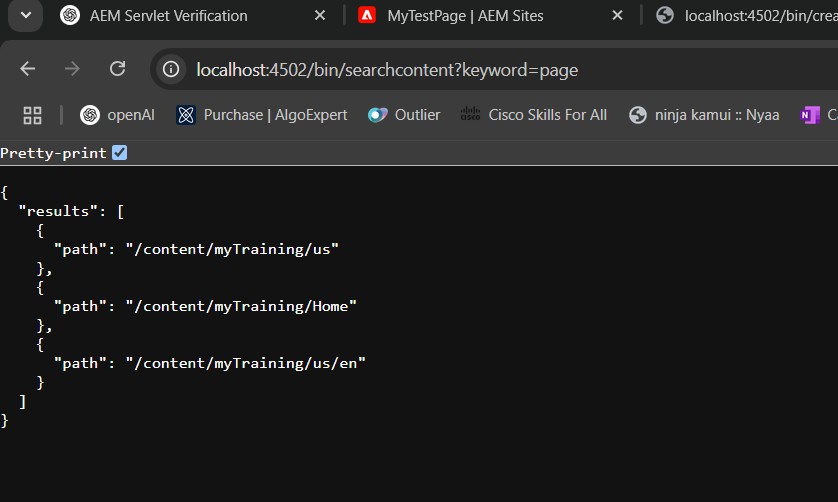
1. **Page Creation Using PageManager API Check Output:** 
   * **Steps:**

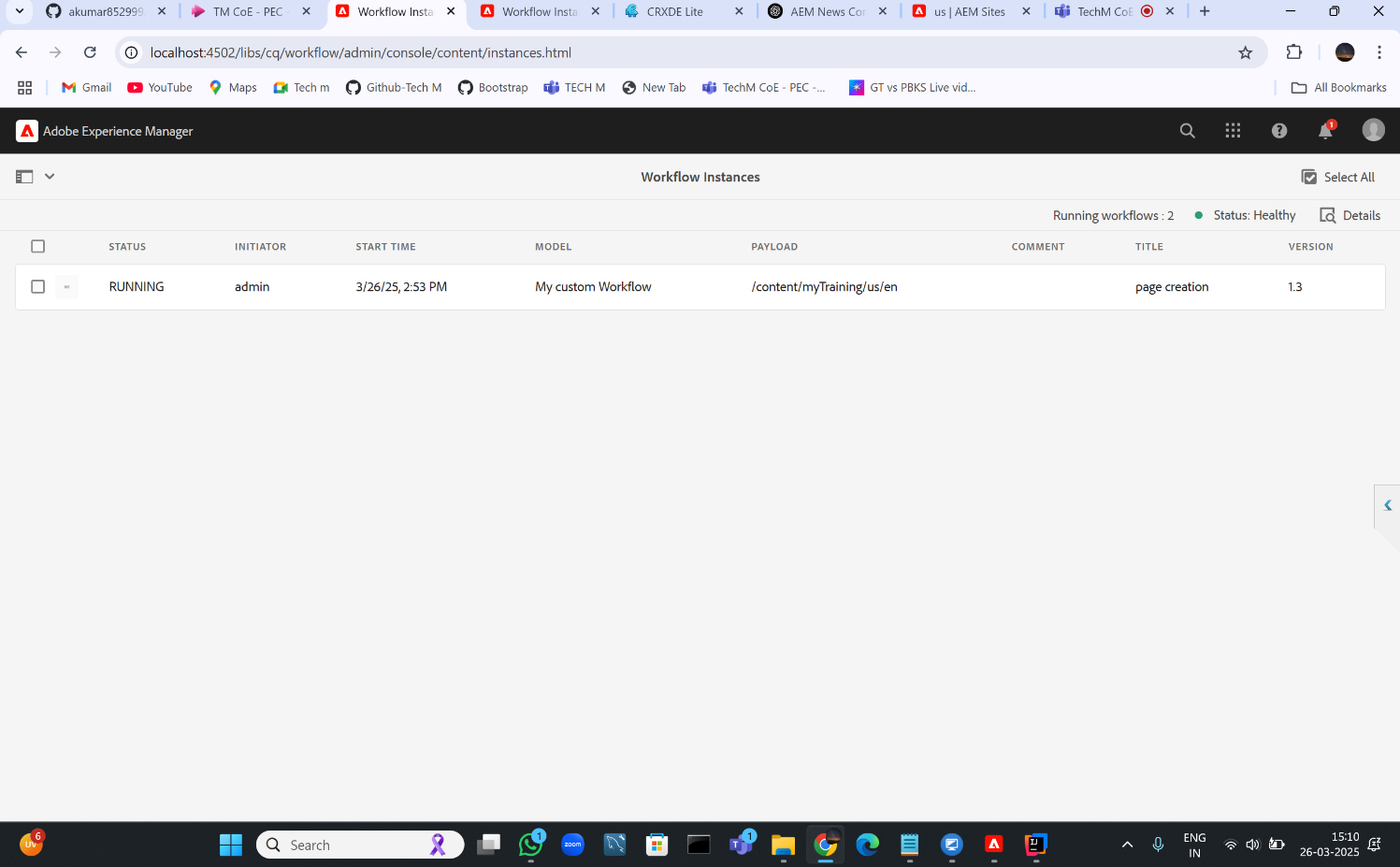
1. Run CreatePageServlet (Step 2).

Page created:



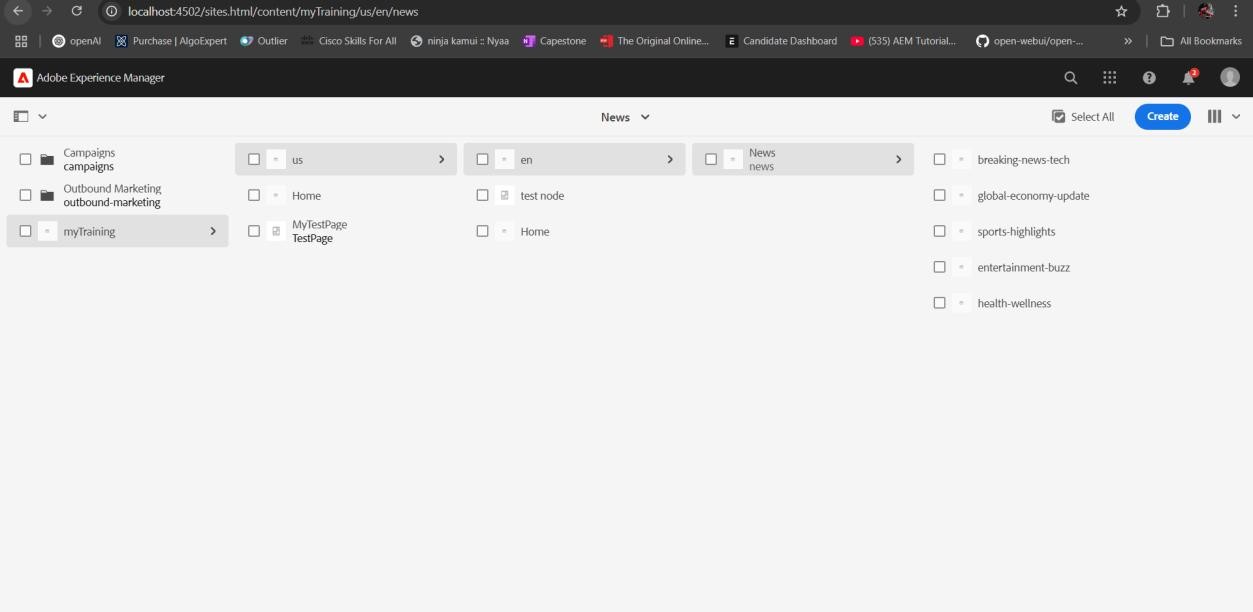
1. SearchServlet (PredicateMap-based search)



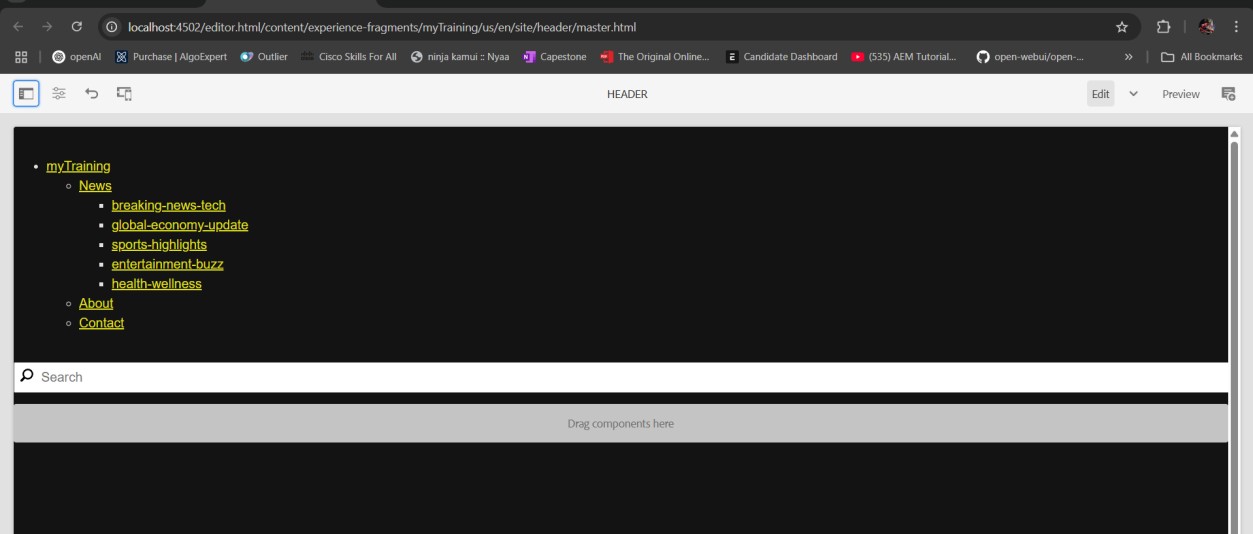


**DATE : 24-03-2025**

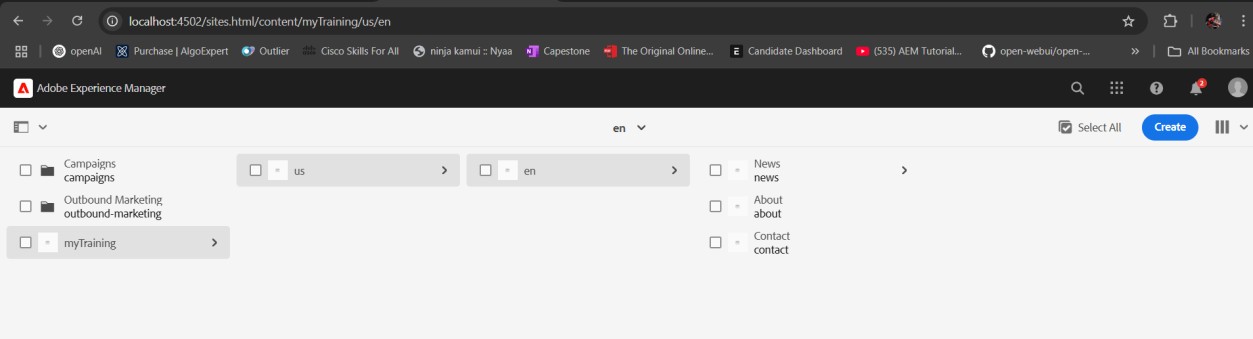
Task 1: Create 5 News Article Pages under /content/us/en/news



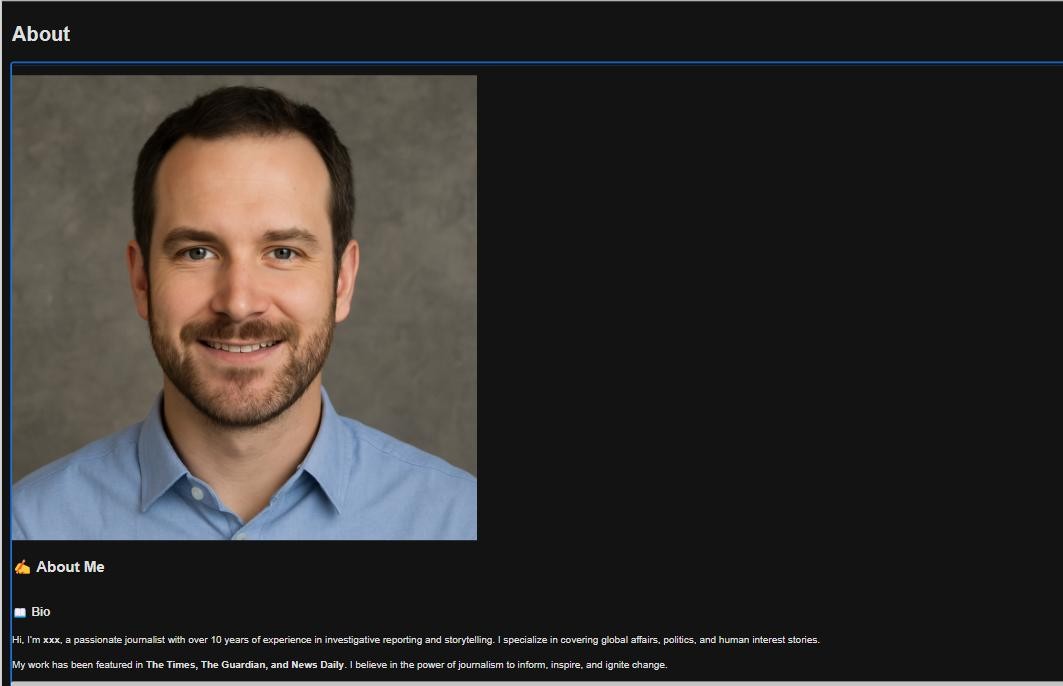
Task 2: Create Header Experience Fragment and Use These Pages as Menu



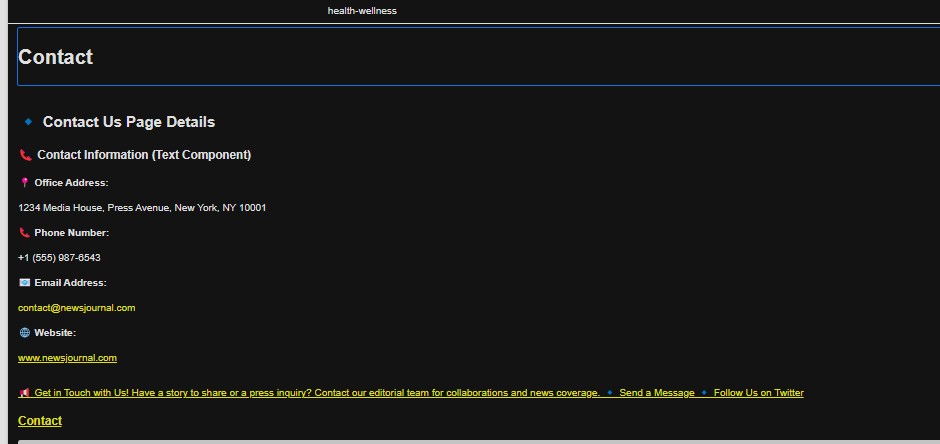
Task 3: Create "Contact Us" and "About Me" Pages



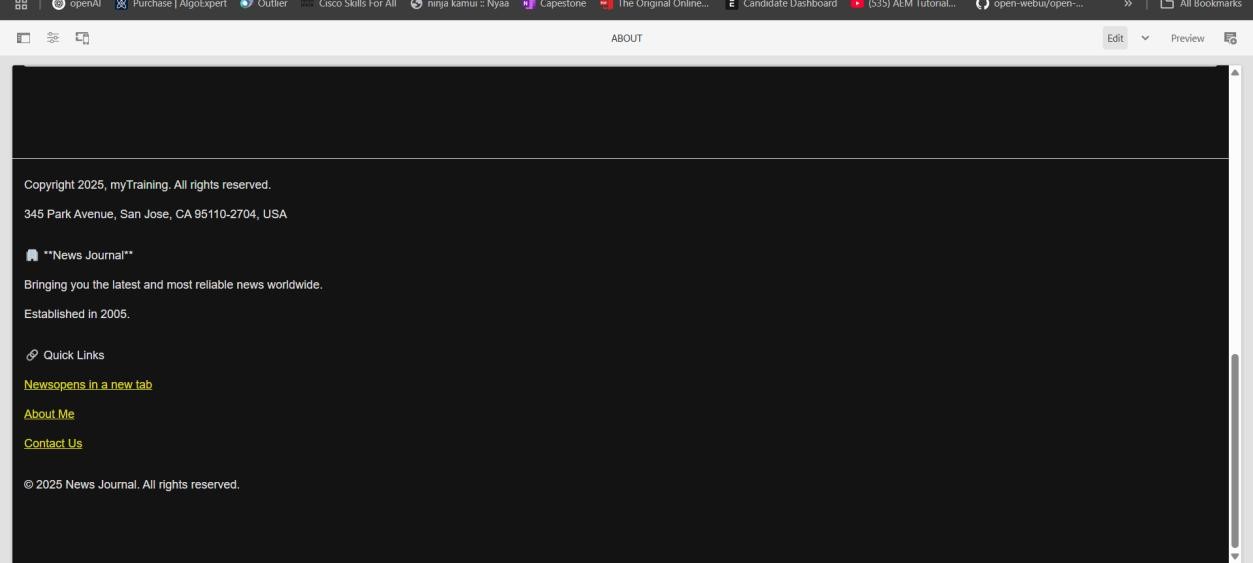
About me page:



Contact page:



Task 4: Create Footer Experience Fragment with 4 Sections



**DATE : 27-03-2025**

Step 3: Create a Site Using MSM with us/en as the Source

