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

DAM (Digital Asset Management) in AEM is a system used for storing, managing, and delivering

digital assets such as images, videos, and documents.

**Why we use it:**

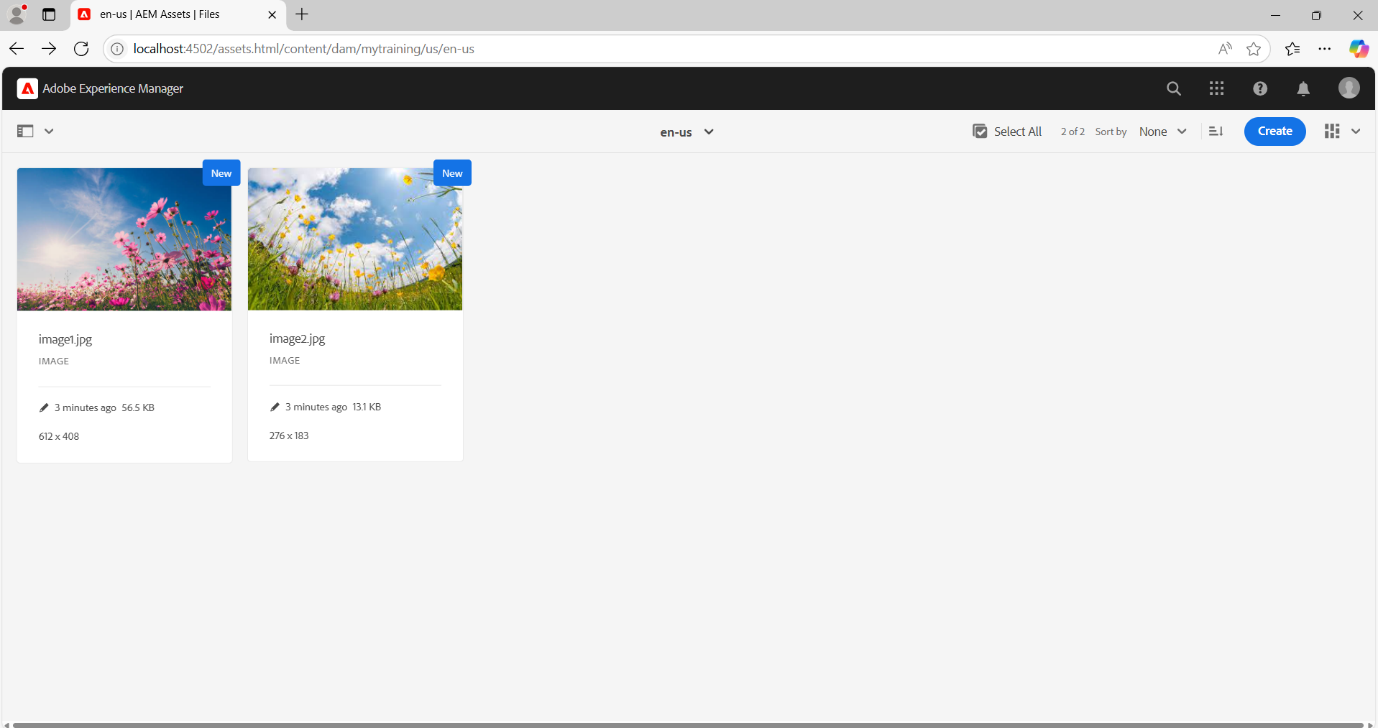
Centralized asset repository for easy access and version control.

Streamlined workflow and collaboration across teams.

Automated rendition generation for different device formats.

Improves content reuse and delivery efficiency.

**2. Create one folder inside our project folder and follow the path: “/content/dam/myTraining/us/en-us” and upload 2 images and author those images on the page using the image component.**

****

**A screenshot of a computer

AI-generated content may be incorrect.**

**3. What is renditions? Check the renditions for 2 images which we had uploaded.**

A Rendition in AEM is an automatically generated variation of an image in different sizes

and resolutions.

Purpose:

Optimizes image delivery based on device type.

Improves performance by serving smaller images to mobile devices.

Allows custom image processing (cropping, scaling, etc.).

**4. Add two fields in hello world component with name FirstName and Last Name and print the values on hello world.html component using properties.**

**5. Give a try using @ValueMapValue annotation on HelloWorldModel sling java model “com.myTraining.core.models.HelloWorldModel”.**

**6. Why we are using package manager and jar? Create package for the hello world component and 2 images what we have uploaded.**

**There should be 2 packages. First one for dam which is images and second one is for helloworld component.**

Package Manager bundles AEM content, code, and configurations into .zip files for easy deployment and migration.

It simplifies exporting, importing, and versioning content across AEM instances.

JAR files package Java classes and resources into a single archive.

They are used for deploying and reusing Java-based services in AEM.

Both streamline deployment, backup, and modularization in AEM.

**7. Configure replication agent and publish the page from author to publish env(4502(author)—>4503 (publish))**