

AEM Assignment 2

1. What is DAM and why do we use it?

DAM (Digital Asset Management)

DAM is a **centralized system** used in AEM to **store, manage, and retrieve digital assets** like images, videos, documents, etc.

Why use DAM?

1. **Centralized Storage** - Keeps all assets in one place.
 2. **Version Control** - Helps track changes to assets.
 3. **Metadata Management** - Allows tagging, categorizing, and searching assets easily.
 4. **Renditions** - Automatically generates different image sizes for optimization.
 5. **Workflow Support** - Automates approvals and publishing processes.
-

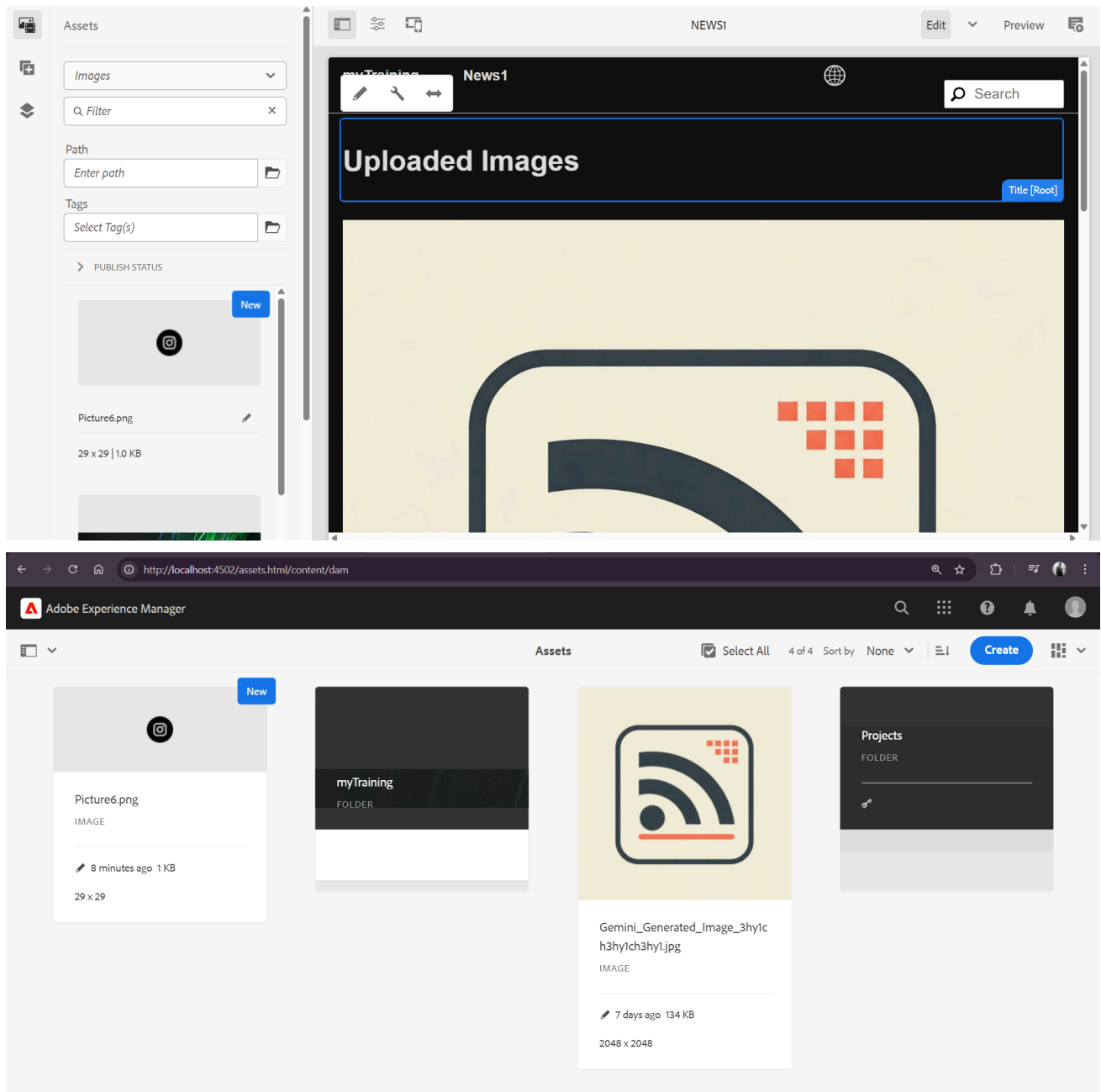
2. Upload Images to DAM & Author Them

Steps:

1. Open **AEM > Navigation > Go to Assets > Files**.
2. Inside DAM, create a new folder:

```
`/content/dam/myTraining/us/en-us`
```

3. Upload **2 images** inside this folder.
4. Go to **Sites > Your Page** (where you want to add images).
5. Edit the page and use the **Image Component**.
6. Drag and drop the images from DAM into the component.
7. Save and publish the page.



3. What is Renditions & How to Check Them?

Renditions

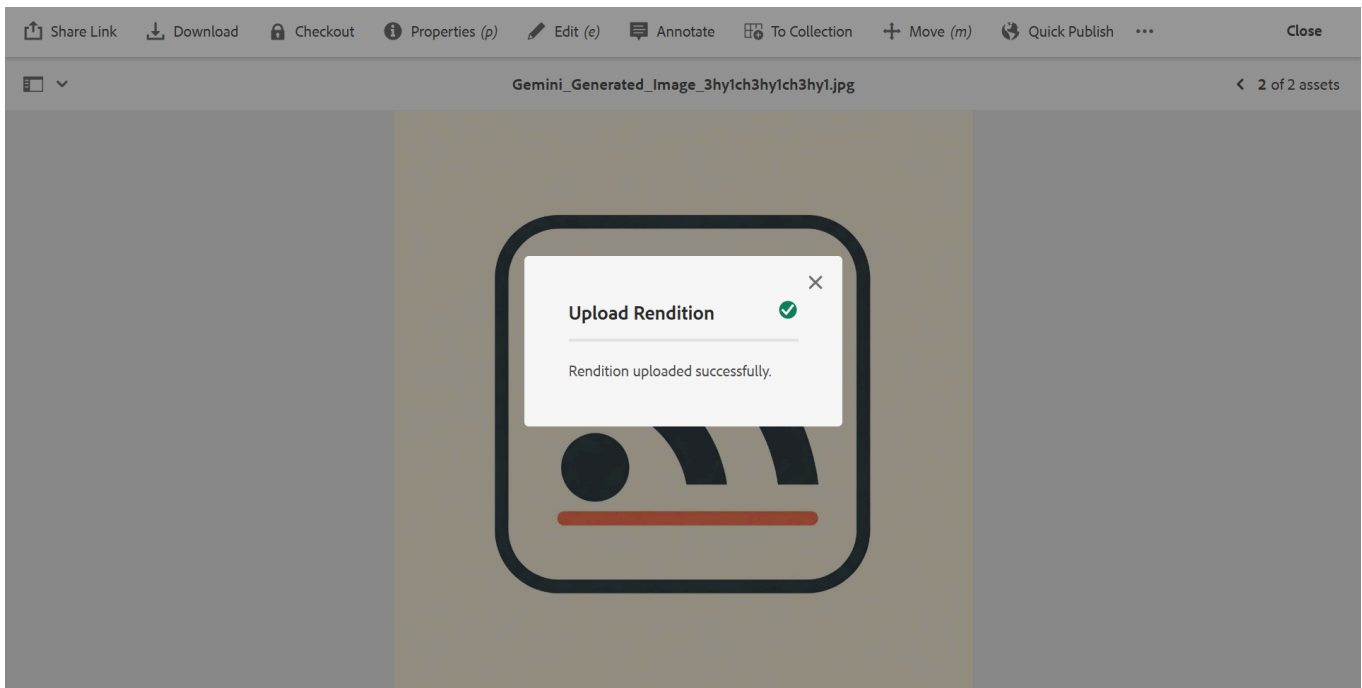
- Renditions are **automatically generated versions** of an uploaded image in different sizes and formats.
- They help optimize images for different devices (e.g., thumbnail, web-optimized versions).

How to Check Renditions?

1. Go to **AEM > Assets > Files**.

2. Open one of the uploaded images.
3. In the **"Properties" panel**, navigate to **"Renditions"**.
4. You'll see different versions like:
 - **Original**
 - **Thumbnail (140x100)**
 - **Web-optimized (1280x720, 600x400, etc.)**

Repeat for the second image.



4. Modify Hello World Component

Steps to Add FirstName & LastName Fields:

1. Go to `ui.apps > components > helloworld` in your AEM project.
2. Open the **dialog.xml** file of the component.
3. Add two text fields for **FirstName** and **LastName**:

...

4. Update Sling Model (HelloWorldModel.java):

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

```
import org.apache.sling.api.resource.Resource;
import org.apache.sling.models.annotations.DefaultInjectionStrategy;
import org.apache.sling.models.annotations.Model;
import org.apache.sling.models.annotations.injectorspecific.ValueMapValue;
```

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

```
    @ValueMapValue
    private String firstName;

    @ValueMapValue
    private String lastName;

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }
}
```

```
}
```

```
```
```

#### 5. Modify helloWorld.html:

```
```html
```

```
<div>
```

```
<h2>First Name: ${properties.firstName}</h2>
```

```
<h2>Last Name: ${properties.lastName}</h2>
```

```
...
```

5. Why Use Package Manager & Create Packages

Why Package Manager?

- It allows us to **export and import content** (e.g., components, images) across AEM instances.
- Helps with **backup, migration, and deployment**.

Create Two Packages:

1. Package 1 (For DAM Images)

- Go to **Tools > Deployment > Package Manager**.
- Click **Create Package** → Name: `DAM-Images-Package`.
- Add filter path:

```
`/content/dam/myTraining/us/en-us`
```

- Save and Build the package.

2. Package 2 (For Hello World Component)

- Create another package: `HelloWorld-Component-Package`.
- Add filter path:

```
`/apps/myTraining/components/helloworld`
```

6. Configure Replication Agent & Publish Page

Steps:

1. **Go to Configuration Manager** in AEM:

`http://localhost:4502/system/console/configMgr`

2. Search for **"Replication Agent"**.
 3. Click **Edit** and set up the replication to **http://localhost:4503** (publish instance).
 4. Test the replication by publishing a page from **author (4502) to publish (4503)**.
-

7. Create News Component using Sling Model

Steps:

1. Create a new component: `news-component` .
2. Add dialog fields for:
 - **Title** (`newsTitle`)
 - **Details** (`newsDetail`)
 - **Published Date** (`newsDate`)
 - **Source** (`newsSource`)
3. Create a **Sling Model** for News:

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

```
import org.apache.sling.api.resource.Resource;
import org.apache.sling.models.annotations.Model;
import org.apache.sling.models.annotations.injectorspecific.ValueMapValue;
```

```
@Model(adaptables = Resource.class)
public class NewsModel {
```

```
    @ValueMapValue
    private String newsTitle;

    @ValueMapValue
    private String newsDetail;

    @ValueMapValue
    private String newsDate;
```

```

@ValueMapValue
private String newsSource;

public String getNewsTitle() {
    return newsTitle;
}

public String getNewsDetail() {
    return newsDetail;
}

public String getNewsDate() {
    return newsDate;
}

public String getNewsSource() {
    return newsSource;
}

}

```

4. ****Modify news-component.html**:**

```

```html

<div>
<h2>${properties.newsTitle}</h2>
<p>${properties.newsDetail}</p>
${properties.newsDate}
${properties.newsSource}
</div>

```

```

5. ****Apply Styles using Clientlibs****

- Create ****`news-clientlib`**** under ``/apps/myTraining/clientlibs/news``.

– Add CSS (`news.css`):

```
``css
h2 { color: green; }
p { color: yellow; }
span { color: black; }
```
