News Component Development Using Sling Model

News Component Overview:

The news component was developed using a Sling Model, enabling dynamic content retrieval. It includes the following fields:

- **News Title** – Rendered as `<h2>` in **green** color.

- **News Detail** – Displayed as `<p>` in **yellow** color.

- **Published Date** – Shown in **black** color.

- **Source** – Identifies the news origin.

The Sling Model ensures that these fields are dynamically fetched and presented within the component.

Multi-Field News Component:

A multi-field structure was implemented to allow content authors to add multiple news entries dynamically. Key features include:

- A **repeatable structure** to accommodate multiple news articles.

- Fields for **News Title** and **Source** for each entry.

This enables authors to manage multiple news articles efficiently within a single component instance.

Clientlibs for News Component:

Client-side libraries (`clientlibs`) were used to apply specific styles to the component:

- **News Title (h2)** – Styled in **green**.

- **News Detail (<p>)** – Appears in **yellow**.

- **Published Date** – Displayed in **black**.

These styles were integrated using AEM's **Clientlibs** mechanism to ensure consistency across the platform.

Custom Component Styling:

A custom style class named **"cop-news-component"** was added, allowing content authors to apply predefined styles effortlessly.

Base Page Component & Metadata Configuration:

A **base page component** was created and linked to a metadata configuration to dynamically include Open Graph (OG) properties. These properties help optimize social media sharing and include:

- **og:title** – Specifies the page title.

- **og:description** – Provides a brief page summary.

- **og:image** – Defines the preview image for social media.

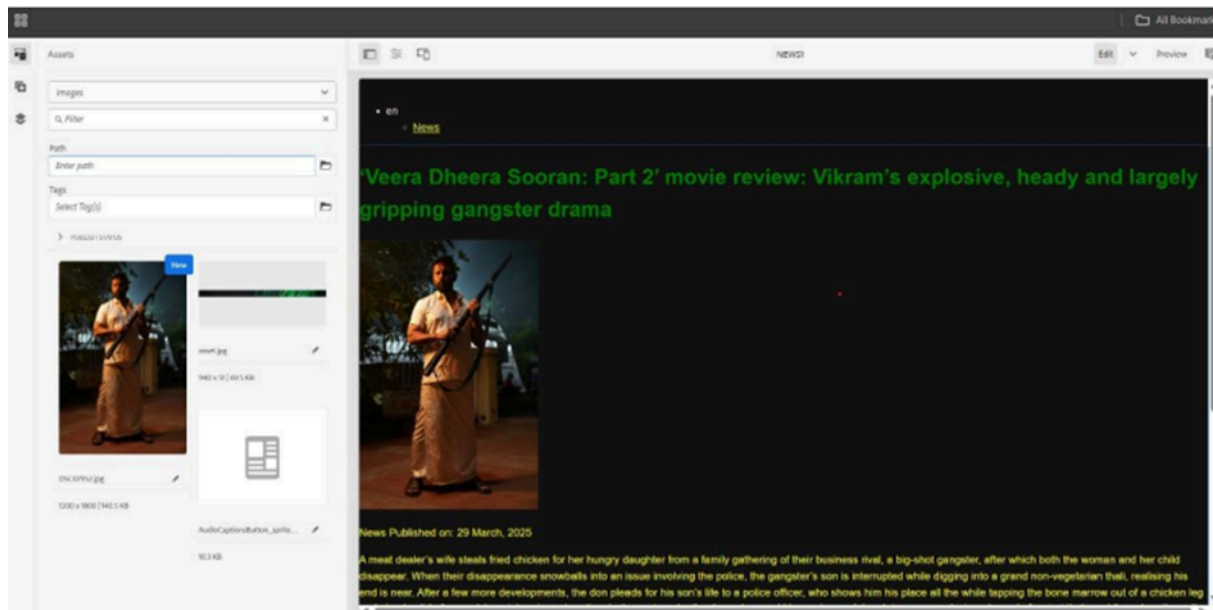This ensures that when a page is shared, appropriate metadata is displayed.

Custom Page Properties:

A **custom page properties** configuration was introduced under **"Global Properties"**, enabling authors to set Open Graph metadata directly from the AEM page properties. The fields include:

- **OG Title**

- **OG Description**

- **OG Image Path**

ExtraClientLibs in Multi-Field Component:

The **extraClientLibs** property was leveraged within the multi-field component to ensure that additional scripts and dependencies are loaded dynamically. This improves flexibility and optimizes the performance of the component.

Code

```java
@Model(adaptables = Resource.class)
public class NewsModel {
@Inject
private String newsHeadline;
@ValueMapValue
private String newsDetail;
@ValueMapValue
private Date newsPublishDate;
@ValueMapValue
private String newsImg;
@ValueMapValue
private List<String> newsTitle;
public String getNewsDetail() {
return newsDetail;
}
public Date getNewsPublishDate() {
return newsPublishDate;
}
public String getNewsImg() {
return newsImg;
}
public String getNewsHeadline() {
return newsHeadline;
}
public List<String> getNewsTitle() {
if (newsTitle != null) {
return new ArrayList<String>(newsTitle);
} else {
return Collections.emptyList();
}
}
```

```
}
}
(news.html):
<div class="news-item">
<h2>${newsModel.newsHeadline}</h2>
<p>${newsModel.newsDetail}</p>
<span>${newsModel.newsPublishDate}</span>
<img src="${newsModel.newsImg}" alt="News Image">
<ul>
<sly data-sly-list.title="${newsModel.newsTitle}">
<li>${title}</li>
</sly>
</ul>
</div>


@Model(adaptables = Resource.class)
public class MultiNewsModel {
@Inject
@Children
private List<NewsModel> newsList;
public List<NewsModel> getNewsList() { return newsList; }
}

div class="multi-news">
<sly data-sly-list.newsItem="${multiNewsModel.newsList}">
<h2>${newsItem.title}</h2>
<p>${newsItem.source}</p>
</sly>
</div>
```