**1. Header Implementation**

**Objective:** Use OOTB Header component and configure it with navigation links.

**Steps:**

1. Open AEM, go to **Templates** and edit your base template (Content Page or Book Page Template).
2. In the **Structure** section of the template, drag and drop the **OOTB Header component** (from Core Components).
3. Style the header using client libraries or style system to ensure proper spacing, font, and hover behavior for dropdown.

**2. Footer Implementation**

**Objective:** Use a custom Footer component or editable templates to include the following:

**Sections:**

* About Me
* Contact Me
* Latest 5 News (with hyperlinks opening in a new tab)
* Social Media section

**Steps:**

1. On the same template, scroll to the bottom and add **OOTB Footer component** or create a custom footer container.
2. Add the following in the Footer layout:
   * **Static Text** for *About Me* and *Contact Me* (linked to respective pages).
   * Create a **Custom News List Component**:
     + Use Sling Model to fetch latest 5 pages from /content/news
     + Display as list of links (<a target="\_blank">) that open in new tabs.
   * Add **Social Media Icons or Links** using Static HTML or Teaser components with links.

**3. Home Page Implementation**

**Objective:** Use Teaser to display a journalist’s award ceremony announcement.

**Steps:**

1. Create a **Home Page** using **Content Page Template** under /content/your-site.
2. Drag and drop **Core Teaser Component** on the page.
3. Configure the Teaser:
   * Add an image of the award ceremony
   * Add title and short description
   * Use CTA (Call To Action) button:
     + Link it to the full news article page about the award (which should be created under /content/news).
4. Style the teaser appropriately using Style System or Design Dialog options.

**4. News Page and NewsRoom Component**

**Objective:** Display all authored news pages as cards using a custom NewsRoom component.

**Steps:**

1. **Create News Page** under /content using **Content Page Template**.
2. **Create a custom AEM component** named **NewsRoom**:
   * In the Sling Model/Java backend:
     + Fetch all child pages under /content/news
     + From each page, get:
       - Page Title
       - Image (from image component)
       - Description (from text component)
   * In the HTL (HTML Template Language):
     + Loop through each fetched page
     + Display as cards: Image, Title, Description
     + Each card should link to the full news article page
3. **Drag and drop NewsRoom component** on the News page and author the parent path as /content/news.

package com.yoursite.core.models;

import com.adobe.cq.sightly.WCMUsePojo;

import org.apache.sling.api.resource.Resource;

import org.apache.sling.api.SlingHttpServletRequest;

import org.apache.sling.api.resource.ValueMap;

import org.apache.sling.models.annotations.DefaultInjectionStrategy;

import org.apache.sling.models.annotations.Model;

import org.apache.sling.models.annotations.injectorspecific.Self;

import org.apache.sling.models.annotations.injectorspecific.Inject;

import java.util.ArrayList;

import java.util.List;

@Model(adaptables = SlingHttpServletRequest.class, defaultInjectionStrategy = DefaultInjectionStrategy.OPTIONAL)

public class NewsDisplayModel {

@Inject

private Resource resource;

@Self

private SlingHttpServletRequest request;

public List<NewsEntry> getEntries() {

List<NewsEntry> list = new ArrayList<>();

Resource parent = request.getResourceResolver().getResource("/content/news");

if (parent != null) {

for (Resource page : parent.getChildren()) {

ValueMap props = page.getValueMap();

String title = props.get("jcr:title", String.class);

String image = props.get("image", String.class);

String text = props.get("text", String.class);

if (title != null && image != null && text != null) {

list.add(new NewsEntry(title, image, text));

}

}

}

return list;

}

// Inner class representing a news entry

public static class NewsEntry {

private final String title;

private final String image;

private final String text;

public NewsEntry(String title, String image, String text) {

this.title = title;

this.image = image;

this.text = text;

}

public String getTitle() {

return title;

}

public String getImage() {

return image;

}

public String getText() {

return text;

}

}

}

HTL:

<sly data-sly-use.model="com.yoursite.core.models.NewsDisplayModel">

<div class="newsroom-cards-container">

<sly data-sly-list.entry="${model.entries}">

<div class="news-card">

<img src="${entry.image}" alt="${entry.title}" class="news-image" />

<div class="news-content">

<h3 class="news-title">${entry.title}</h3>

<p class="news-text">${entry.text}</p>

</div>

</div>

</sly>

</div>

</sly>

**5. Create News Article Pages**

**Objective:** Author multiple news pages with unique content under /content/news.

**Steps:**

1. Under /content/news, create several **Content Pages** (e.g., award-news, event-coverage).
2. On each page:
   * Add a **Title** (Page Properties)
   * Add an **Image Component** (author with relevant image)
   * Add a **Text Component** (with news content)
3. Ensure each news article has unique:
   * Title
   * Image
   * Description text

These will be dynamically pulled by the **NewsRoom** component on the News page.

**6. About Me Page – Journalist Component**

**Objective:** Provide journalist info using custom Journalist component.

**Steps:**

1. Create **About Me Page** using **Content Page Template**.
2. Add custom **Journalist Component**:
   * Static content or fields for journalist bio, picture, experience, etc.
3. Ensure only this component is used on this page.
4. Style the layout for a clean professional look.

**7. Contact Me Page**

**Objective:** Provide contact and social media info using OOTB components.

**Steps:**

1. Create **Contact Me Page** using **Content Page Template**.
2. Add either:
   * **Text Component** for email, phone number
   * OR
   * **Teaser Component** for social media links (e.g., LinkedIn, Twitter)
3. Configure teaser with:
   * Icon/Image
   * Title (“Follow Me on Twitter” etc.)
   * Link to the social media profile





