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

For this part, we need to create **five unique news article pages** and place them under /content/us/en/news. We'll use the **News component** you created previously to display the title, news detail, and published date.

Steps:

1. **Create the News Article Pages** (Manually via AEM UI or using JCR)

For each of the five articles, you will use your **News component** to populate the page with:

- **Title**
- **Detail**
- **Published Date**

Page Structure:

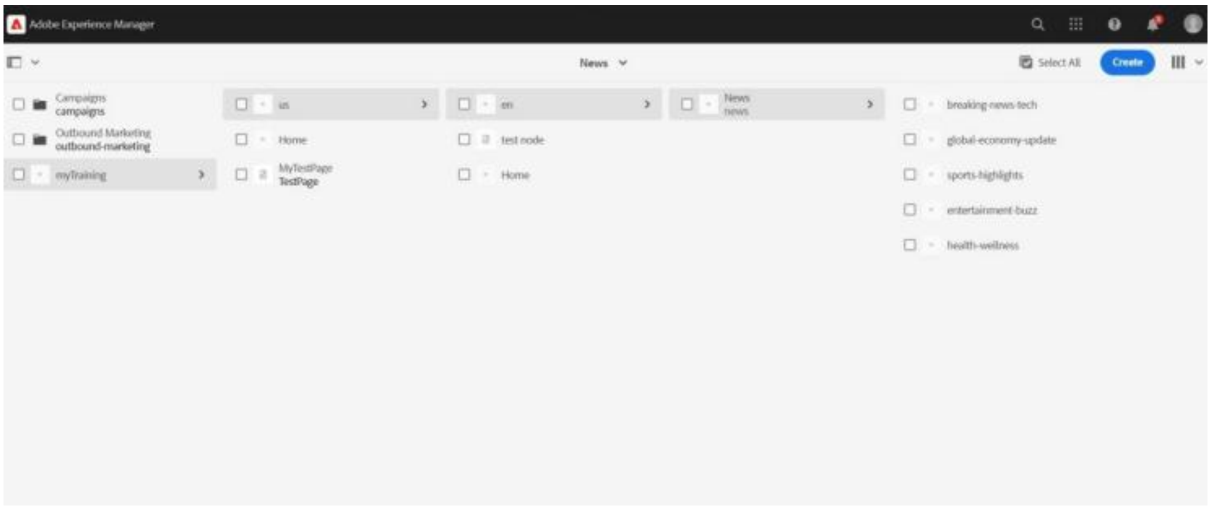
Path: /content/us/en/news/news-article-1 (repeat for news-article-2, news-article-3, etc.)

Each page will have:

- **Title**: Set using the **News component**.
- **Detail**: News content (text or rich text).
- **Published Date**: From sling model.

Structure:

1. /content/us/en/news/news-article-1
2. /content/us/en/news/news-article-2
3. /content/us/en/news/news-article-3
4. /content/us/en/news/news-article-4
5. /content/us/en/news/news-article-5



2. Create Header Experience Fragment for Header and Use Pages as Menu

Create Experience Fragment (Header XF):

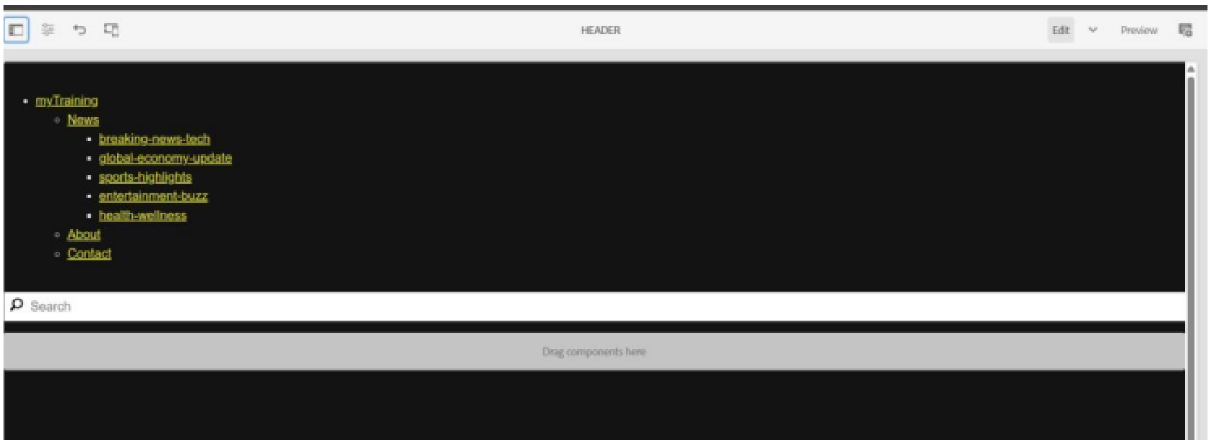
- Path: /content/experience-fragments/us/en/header

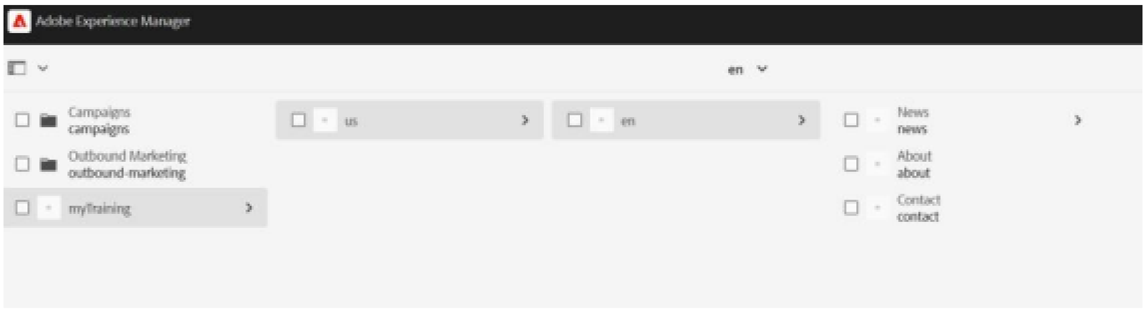
In this fragment, add a **menu** with links to the 5 **news article pages**.

Add Menu Links:
The **XF** will include links to your news articles as a navigation menu.

Example:

- **Menu Item**: "News"
- Under the News menu, you can list the 5 news articles (links to the respective pages)





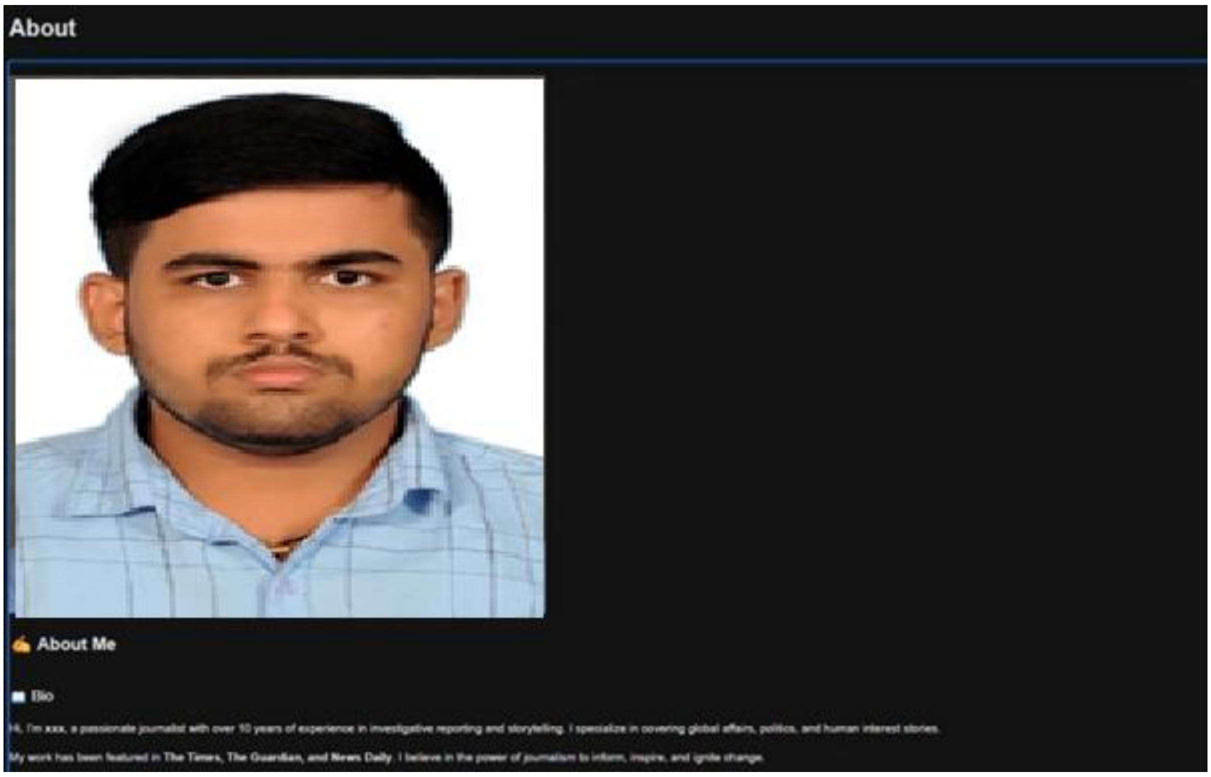
3. Create Contact Us Page and About Me Page

About Me Page: This will contain details about the journalist. Use the **Teaser**, **Image**, **Text**, and **Title** components to populate it.

Contact Us Page: Display contact details like the **mobile number**, **office address**, and **email address**.

Steps:

1. **Create "About Me" Page**:
   Path: /content/us/en/about-me
   - Use **Image** component for a photo.
   - Use **Text** component for content (journalist biography).
   - Use **Teaser** or **Title** for sections.
2. **Create "Contact Us" Page**:
   Path: /content/us/en/contact-us
   - Use **Text** component to list contact details.
   - Include **mobile number**, **office address**, **email address**.



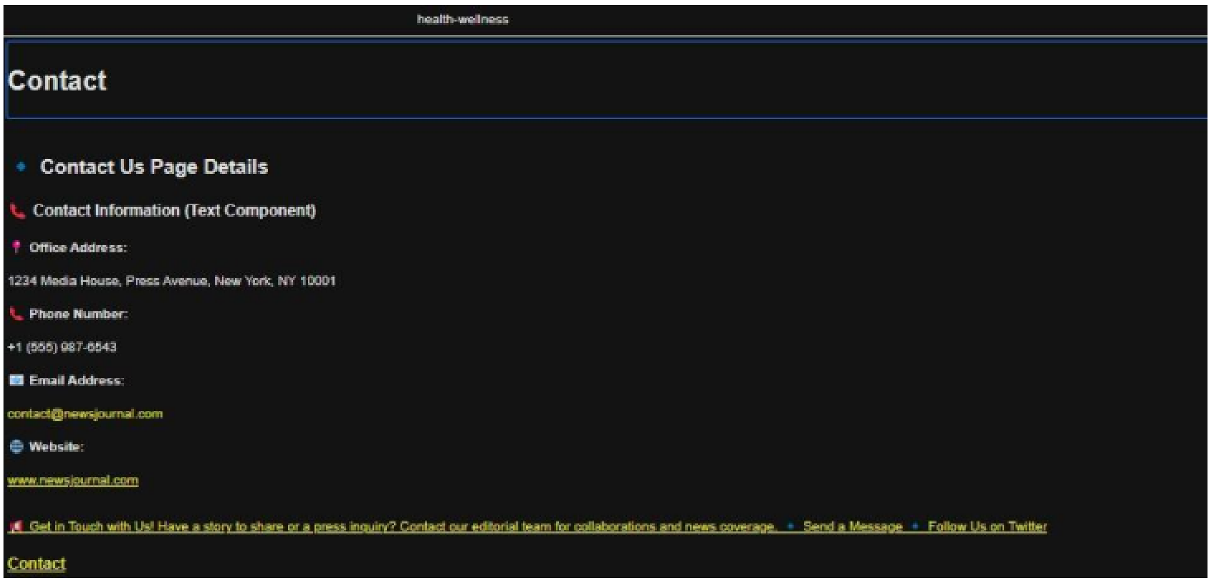
About

About Me

Bio

Hi, I'm xxx, a passionate journalist with over 10 years of experience in investigative reporting and storytelling. I specialize in covering global affairs, politics, and human interest stories.

My work has been featured in The Times, The Guardian, and News Daily. I believe in the power of journalism to inform, inspire, and ignite change.



health-wellness

Contact

Contact Us Page Details

Contact Information (Text Component)

Office Address:

1234 Media House, Press Avenue, New York, NY 10001

Phone Number:

+1 (555) 987-6543

Email Address:

contact@newsjournal.com

Website:

www.newsjournal.com

Get in Touch with Us! Have a story to share or a press inquiry? Contact our editorial team for collaborations and news coverage. • Send a Message • Follow Us on Twitter

Contact

4. Create Footer XF with 4 Sections

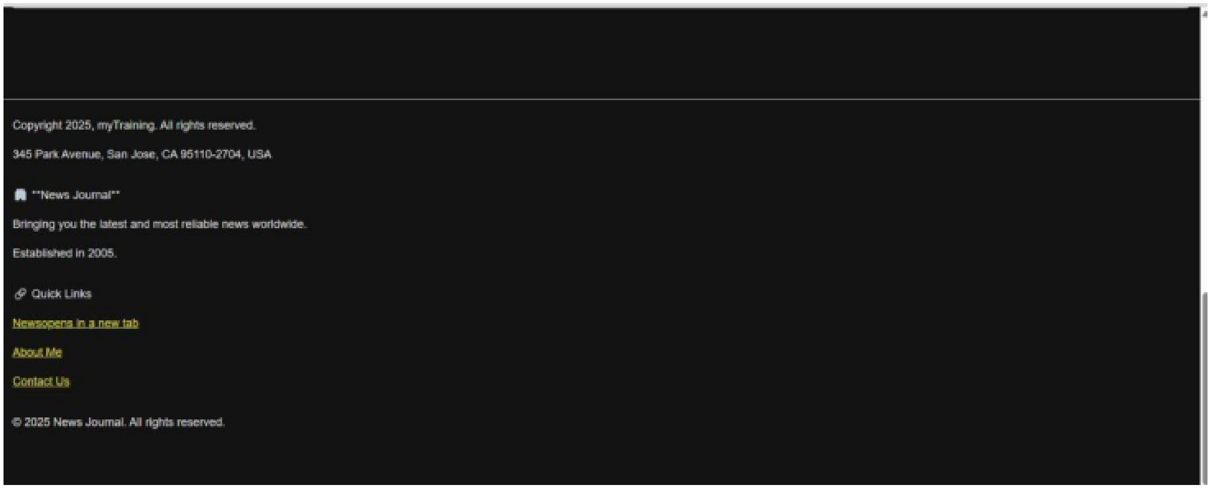
Steps:

1. **Create Footer XF**:
   Path: /content/experience-fragments/us/en/footer

2. **Configure Sections**:

   - **News Menu Section**: Use a **List Component** to display 4 news articles.
   - **About Me Section**: Use **Text Component** for content about the journalist.

- **Contact Us Section**: Use **Text Component** for contact details.
- **Social Media Section**: Use **List Component** to display social media links.



Copyright 2025, myTraining. All rights reserved.

345 Park Avenue, San Jose, CA 95110-2704, USA

News Journal

Bringing you the latest and most reliable news worldwide.

Established in 2005.

Quick Links

Newsopens in a new tab

About Me

Contact Us

© 2025 News Journal. All rights reserved.

5. Create a Custom Service to Print Hello World and Call this Service from News Component Sling Model

1. **Create the Service**:
   Path: /apps/newsroom/core/services/HelloWorldService.java

```
package com.newsroom.core.services;


public interface HelloWorldService {

   String getHelloWorld();

}
```

2. **Implement the Service**:
   Path: /apps/newsroom/core/services/HelloWorldServiceImpl.java

```
package com.newsroom.core.services;


import org.osgi.service.component.annotations.Component;


@Component(service = HelloWorldService.class)

public class HelloWorldServiceImpl implements HelloWorldService {
    @Override
    public String getHelloWorld() {
        return "Hello World from Newsroom Service!";
    }
}
```

3. **Call the Service in News Component Sling Model**:
   Path: /apps/newsroom/core/models/NewsItemModel.java

```
package com.newsroom.core.models;

import com.newsroom.core.services.HelloWorldService;
import org.apache.sling.api.resource.Resource;
import org.apache.sling.models.annotations.Model;

import javax.inject.Inject;

@Model(adaptables = Resource.class)
public class NewsItemModel {

    @Inject
    private HelloWorldService helloWorldService;

    public void logHelloWorld() {
        System.out.println(helloWorldService.getHelloWorld());
    }
}
```

6 Create a custom service to print hello world and call this service from news component sling model and print this value in logs as well.

1. Create the Service Interface

First, we define the service interface which will provide a method for returning "Hello World".

File: /apps/newsroom/core/services/HelloWorldService.java

```
package com.newsroom.core.services;


public interface HelloWorldService {
    String getHelloWorld();
}
```

- **Explanation**: This interface declares a method getHelloWorld() that returns a String.

2. Implement the Service

Now, we provide the implementation for this service where we will return "Hello World".

File: /apps/newsroom/core/services/HelloWorldServiceImpl.java

```
package com.newsroom.core.services;


import org.osgi.service.component.annotations.Component;


@Component(service = HelloWorldService.class)
public class HelloWorldServiceImpl implements HelloWorldService {


    @Override
    public String getHelloWorld() {
        return "Hello World from Newsroom Service!";
    }
}
```

- **Explanation**:
  - We use the **@Component** annotation to register this class as an **OSGi service** in AEM.
  - This class implements the **HelloWorldService** interface and provides the implementation for the getHelloWorld() method, which returns the string "Hello World from Newsroom Service!".

3. Call the Service from News Component Sling Model

Now, we modify the **News Component Sling Model** to inject and call the custom service. We will also log the result of the service using System.out.println().

File: /apps/newsroom/core/models/NewsItemModel.java

```
package com.newsroom.core.models;


import com.newsroom.core.services.HelloWorldService;

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

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

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;


import javax.inject.Inject;


@Model(adaptables = Resource.class)

public class NewsItemModel {


    private static final Logger LOG = LoggerFactory.getLogger(NewsItemModel.class);


    @Inject
    private HelloWorldService helloWorldService;

    // Method to call the service and log the result
    public void logHelloWorld() {
        String message = helloWorldService.getHelloWorld();
        LOG.info(message); // Logs the message
    }
}
```

- **Explanation**:
    - The **HelloWorldService** is injected into the NewsItemModel using **@Inject**.
    - The **logHelloWorld()** method calls the getHelloWorld() method of the service and logs the returned string using **SLF4J** (LOG.info()).
    - We use **SLF4J** (Simple Logging Facade for Java) to log the message in a proper way, which is the standard logging framework in AEM.

The log will appear in the AEM **error.log** file (or wherever logs are configured for your AEM instance).

4. Call the logHelloWorld() Method in HTL (HTML Template Language)

Finally, you need to call the logHelloWorld() method from the Sling Model in your **HTL** template (HTML).

File: /apps/newsroom/components/news/news-item.html

```
<data-sly-use.newsModel="com.newsroom.core.models.NewsItemModel" />

<!-- Call the method to log Hello World -->
<sly data-sly-call="${newsModel.logHelloWorld}" />
```

- **Explanation**:
    - The **data-sly-use** tag is used to create a reference to the **NewsItemModel** class in the HTL file.
    - **data-sly-call** calls the logHelloWorld() method from the model, which will print "Hello World from Newsroom Service!" in the **AEM logs**.

5. Check the Logs

Once everything is set up, the output "Hello World from Newsroom Service!" will be logged in AEM's **error.log** or **request.log** files.

You can find these logs in the crx-quickstart/logs/ directory of your AEM instance, specifically in:

crx-quickstart/logs/error.log

7. Create Custom Configurations for Third-Party API

Steps:

1. **Create Configuration Interface**:
   Path: /apps/newsroom/core/config/ThirdPartyApiConfig.java

```
package com.newsroom.core.config;


public interface ThirdPartyApiConfig {

   String getApiUrl();

}
```

2. **Create Configuration Implementation**:
   Path: /apps/newsroom/core/config/ThirdPartyApiConfigImpl.java

```
package com.newsroom.core.config;


import org.osgi.service.component.annotations.Component;

import org.osgi.service.metatype.annotations.Designate;


@Designate(ocd = ThirdPartyApiConfigImpl.class)

@Component(service = ThirdPartyApiConfig.class)

public class ThirdPartyApiConfigImpl implements ThirdPartyApiConfig {
    private String apiUrl;


    @Override
    public String getApiUrl() {
        return apiUrl;
    }


    // Bind method to inject the config
    @Activate
    @Modified
    public void activate(String apiUrl) {
        this.apiUrl = apiUrl;
    }
}
```

3. **Call the API and Print in Logs**:
   Modify the **Sling Model** to fetch and print the data.

```
package com.newsroom.core.models;


import com.newsroom.core.config.ThirdPartyApiConfig;
import org.apache.sling.api.resource.Resource;
import org.apache.sling.models.annotations.Model;


import javax.inject.Inject;


@Model(adaptables = Resource.class)
public class NewsItemModel {


    @Inject
    private ThirdPartyApiConfig thirdPartyApiConfig;


    public void fetchApiDataAndLog() {
        String apiUrl = thirdPartyApiConfig.getApiUrl();
        // Code to call the API and log the response (e.g., using HttpClient or any HTTP library)
        System.out.println("Fetching data from API: " + apiUrl);
    }
}
```