

AEM assignment

Create 5 news article pages under /content/us/en/news All should be unique
Use News component which we had created previously to provide the title, news detail and published date.

Step 1: Create 5 News Article Pages

Navigate to AEM Sites → /content/us/en/news.

Click Create → Page, choose the News Template, and enter details:

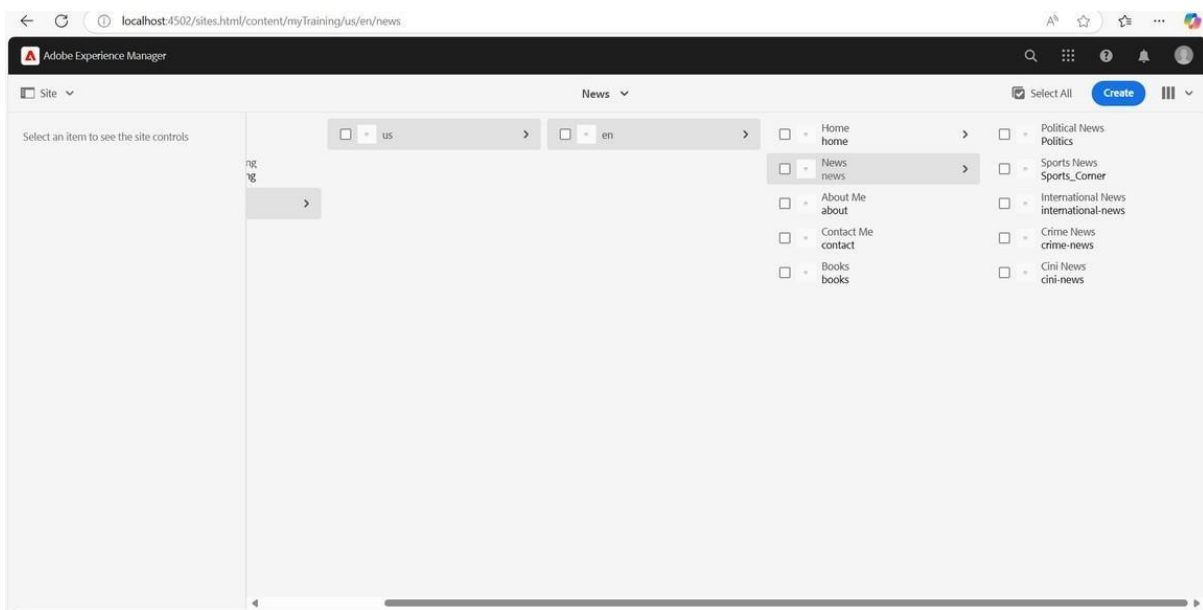
Title: Unique for each article (e.g., "Tech Innovations 2025", "Global Economy Update", "AI in Healthcare", etc.).

Page Name: news-1, news-2, etc.

Open each page and add the News Component:

Title: Set appropriate news title. o News Details: Add content.

Published Date: Set date.



Step 2: Create Header Experience Fragment (XF)

Navigate to AEM → Experience Fragments.

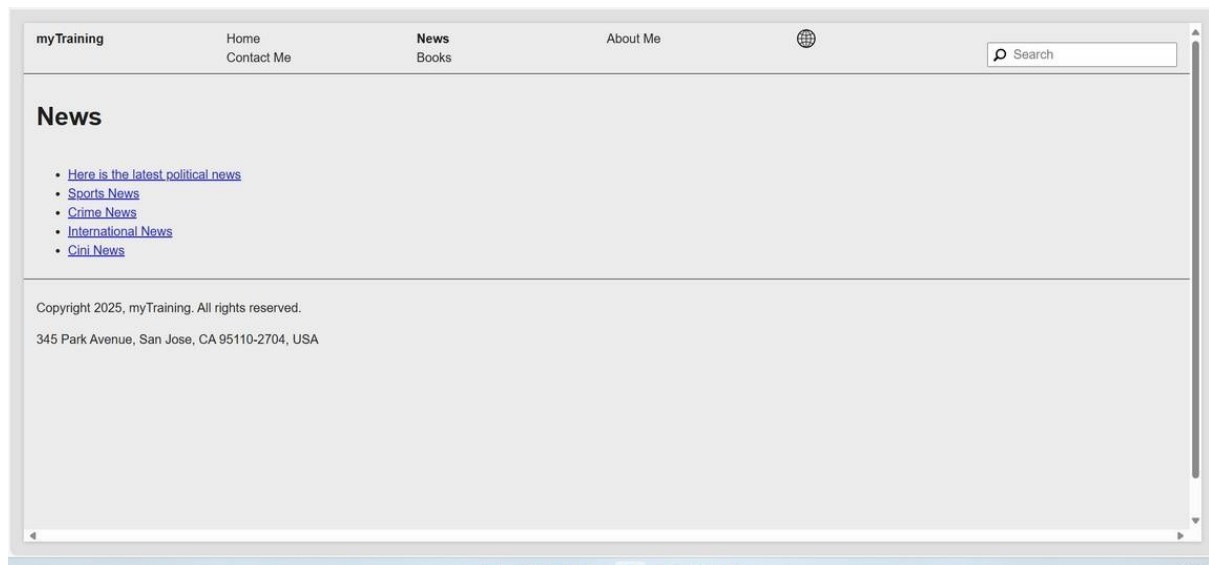
Click Create → Experience Fragment.

Choose Header Template, set the name as News Header, and save it at /content/experience-fragments/us/en/news-header.

Open the News Header XF, add:

o Navigation Component (links to "News", "About Me", and "Contact Us").

Save and publish.



Step 3: Create "About Me" and "Contact Us" Pages

Navigate to /content/us/en/ → Create Page for:

About Me Page o Contact Us Page

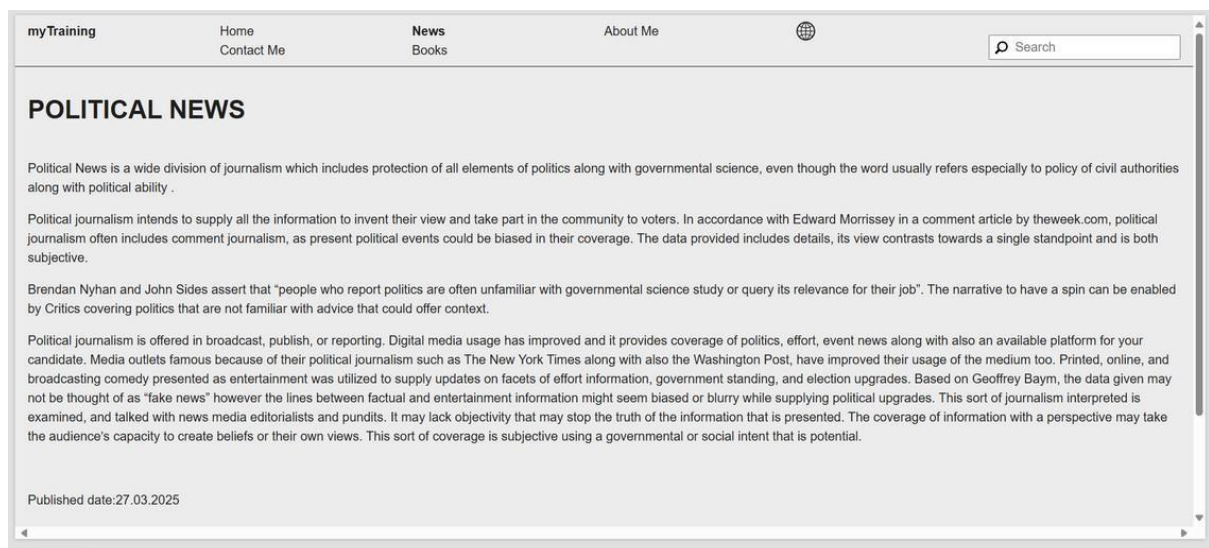
Edit "About Me" Page:

Image Component: Journalist photo. o Text Component: Biography.

Title Component: "About Me".

Edit "Contact Us" Page: o Text Component: Phone number, email, and office address.

Save and publish.



Step 4: Create Footer Experience Fragment (XF)

Navigate to AEM → Experience Fragments.

Click Create → Experience Fragment and choose Footer Template.

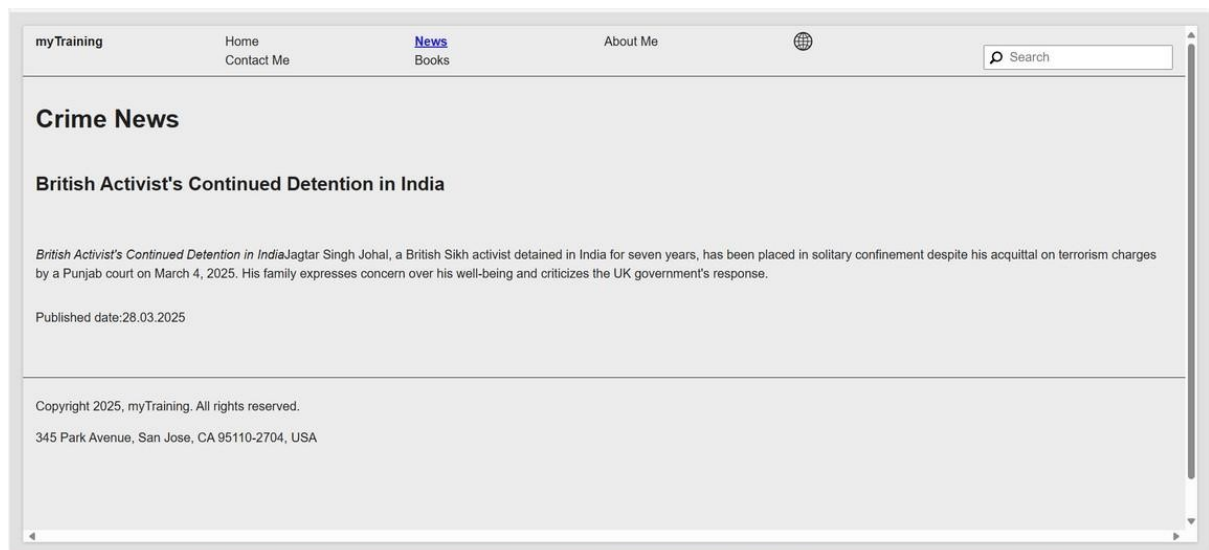
Name it News Footer, path: /content/experience-fragments/us/en/news- footer.

Add sections:

News Menu (List Component): Show 4 recent articles.

About Me (Text Component): Add journalist info. o Contact Us (Text Component): Add contact details.

Social Media (List Component): Add



Step 5: Create a Custom Service to Print "Hello World"

Navigate to your AEM Core Module and create a new Java service:

Java Code:

```
@Component(service = HelloWorldService.class, immediate = true) public class  
HelloWorldService { public String getMessage() {
```

```
return "Hello, World!";
```

```
}
```

```
}
```

Inject this service into your News Component's Sling Model:

Java Code:

```
@Model(adaptables = Resource.class, defaultInjectionStrategy =  
DefaultInjectionStrategy.OPTIONAL) public class NewsModel { @OSGiService private  
HelloWorldService helloWorldService; public String getHelloMessage() {
```

```
return helloWorldService.getMessage();
```

```
}
```

```
}
```

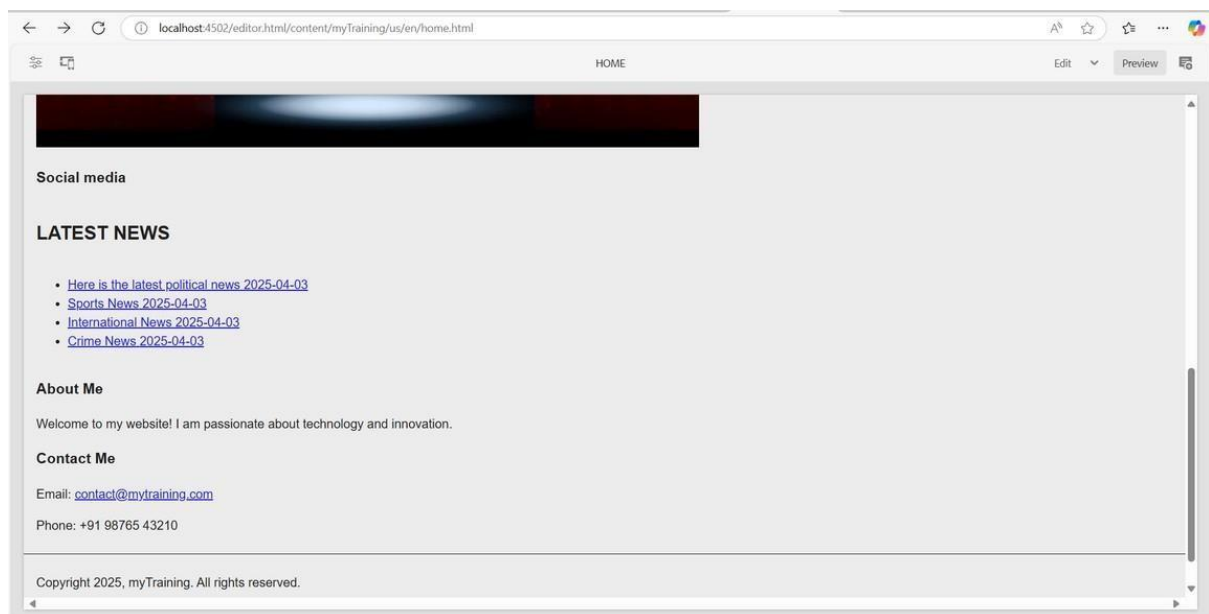
Print the message in logs:

Java Code:

```
private static final Logger LOG = LoggerFactory.getLogger(NewsModel.class); public String
getHelloMessage() {
```

```
String message = helloWorldService.getMessage(); LOG.info("HelloWorldService Output: {}",
message); return message;
```

```
}
```



Step 6: Create Custom Configurations for Third-Party API

Go to AEM → Tools → Configuration Browser.

Click Create → Configuration and name it thirdPartyAPIConfig.

Open OSGi Configuration Console and create a new configuration:

Java code:

```
@Component(service = ThirdPartyAPIService.class, immediate = true)
```

```
@Designate(ocd = ThirdPartyAPIConfig.class)
```

```
public class ThirdPartyAPIService {
```

```
@Activate @Modified
```

```
protected void activate(ThirdPartyAPIConfig config) { apiUrl = config.apiUrl();
```

```
}
```

```
public String fetchAPIData() { try (CloseableHttpClient client = HttpClients.createDefault()) {
```

```
HttpGet request = new HttpGet(apiUrl); HttpResponse response = client.execute(request);
return EntityUtils.toString(response.getEntity());
} catch (Exception e) {
LOG.error("Error fetching API data", e); return "{}";
}
}
}
```

```
@ObjectClassDefinition(name = "Third-Party API Configuration") public @interface
ThirdPartyAPIConfig {
@AttributeDefinition(name = "API URL", description = "Enter the API URL")
String apiUrl() default "https://jsonplaceholder.typicode.com/posts";
}
```

4. Deploy and check logs (tail -f error.log)

o Expected Output: JSON data from <https://jsonplaceholder.typicode.com/posts>.