# AEM News Article Pages & Custom Implementations

## 1. Create 5 News Article Pages

Path: /content/us/en/news

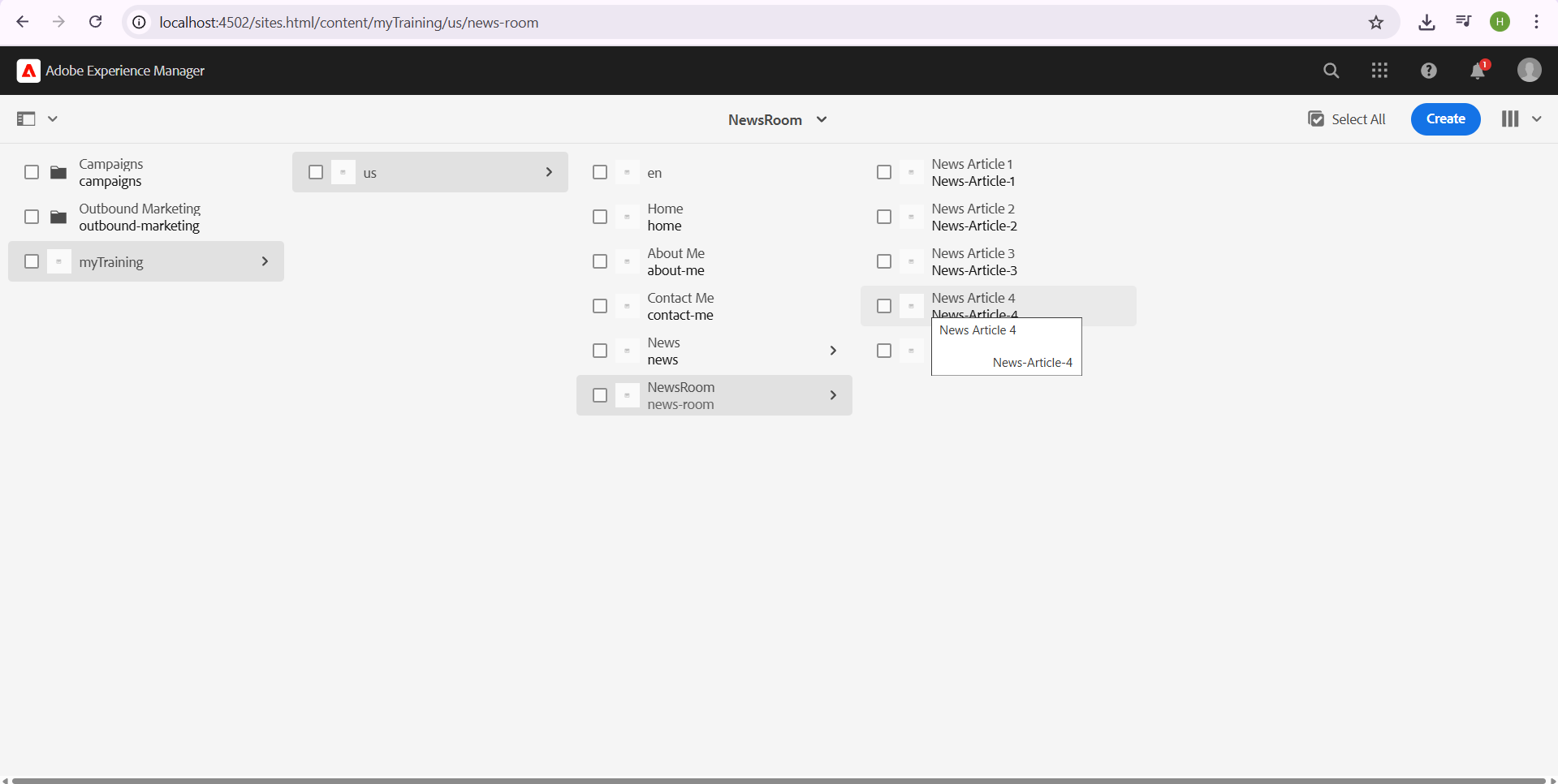
• Create five unique news articles.

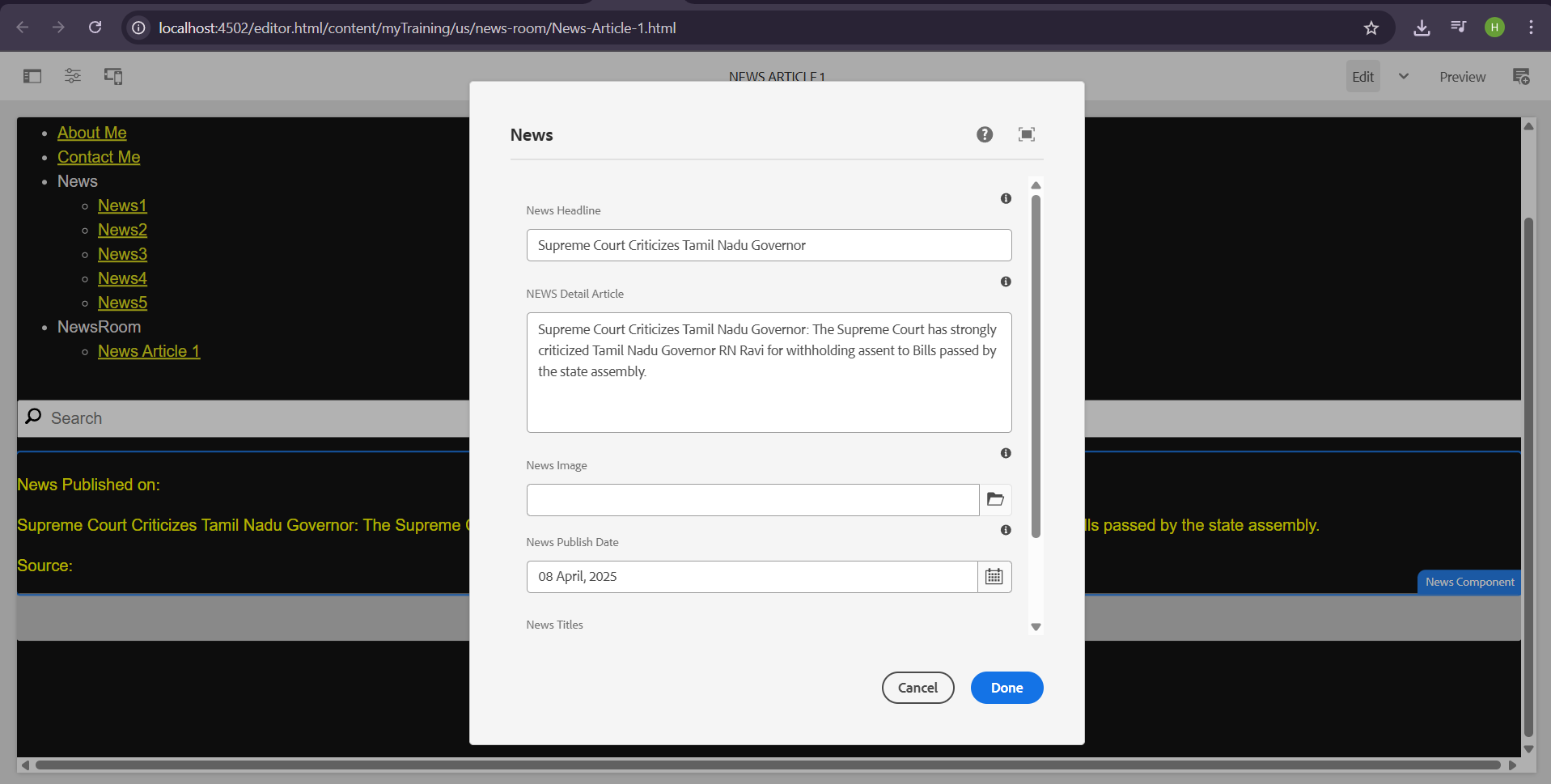
• Use the previously created News Component to provide:

- Title  
 - News Detail  
 - Published Date

1. **Go to AEM Sites**:  
   <http://localhost:4502/sites.html>
2. Navigate to:  
   /content/us/en/
3. Create a new page:

* Template: **News Room Template** (created earlier)
* Page Title: News Article 1  
  Repeat for 5 pages:  
  News Article 1 to News Article 5



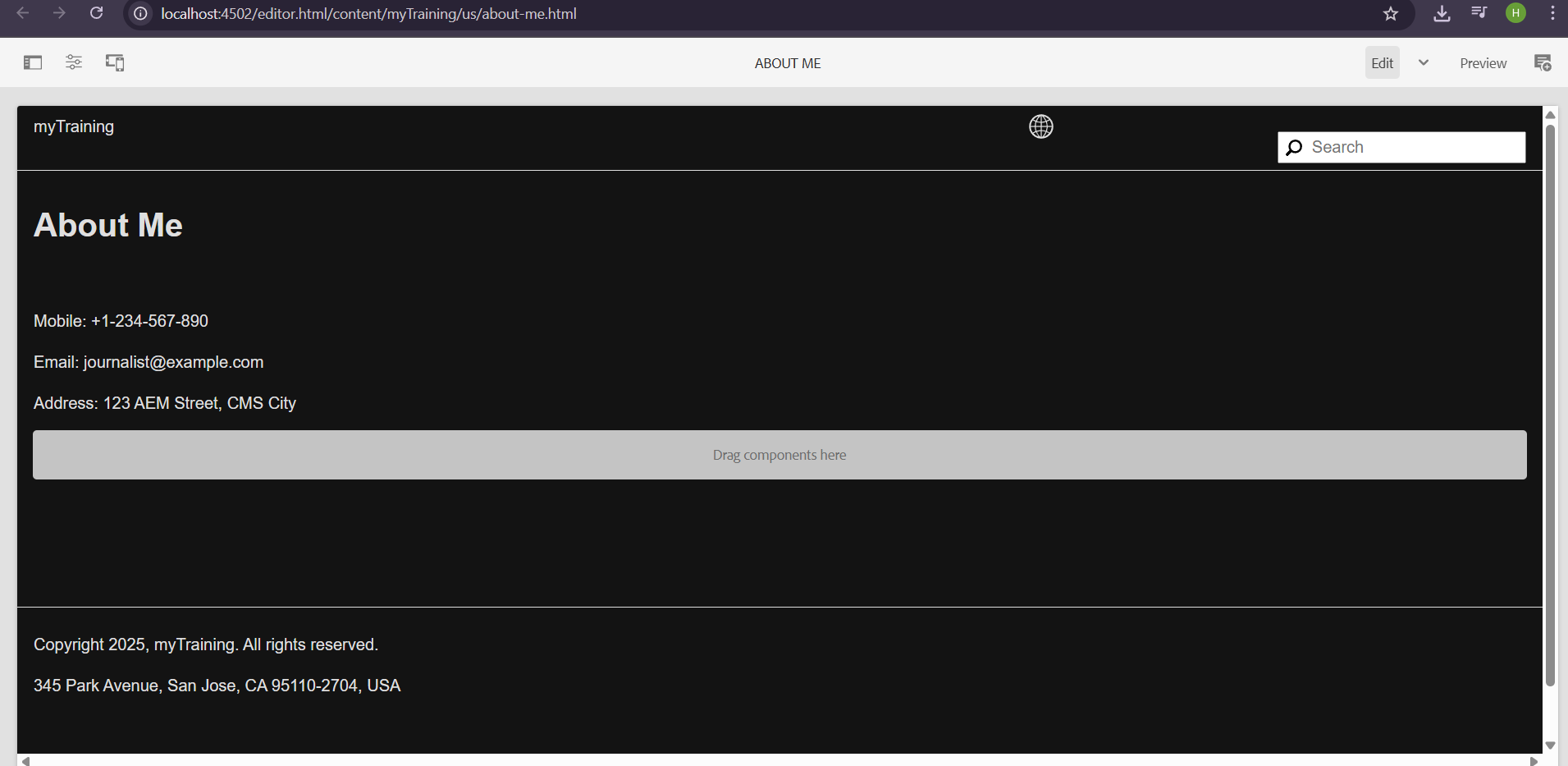


## 2. Create Header Experience Fragment (XF)

• Create an Experience Fragment for the Header.

• Add a navigation menu with:

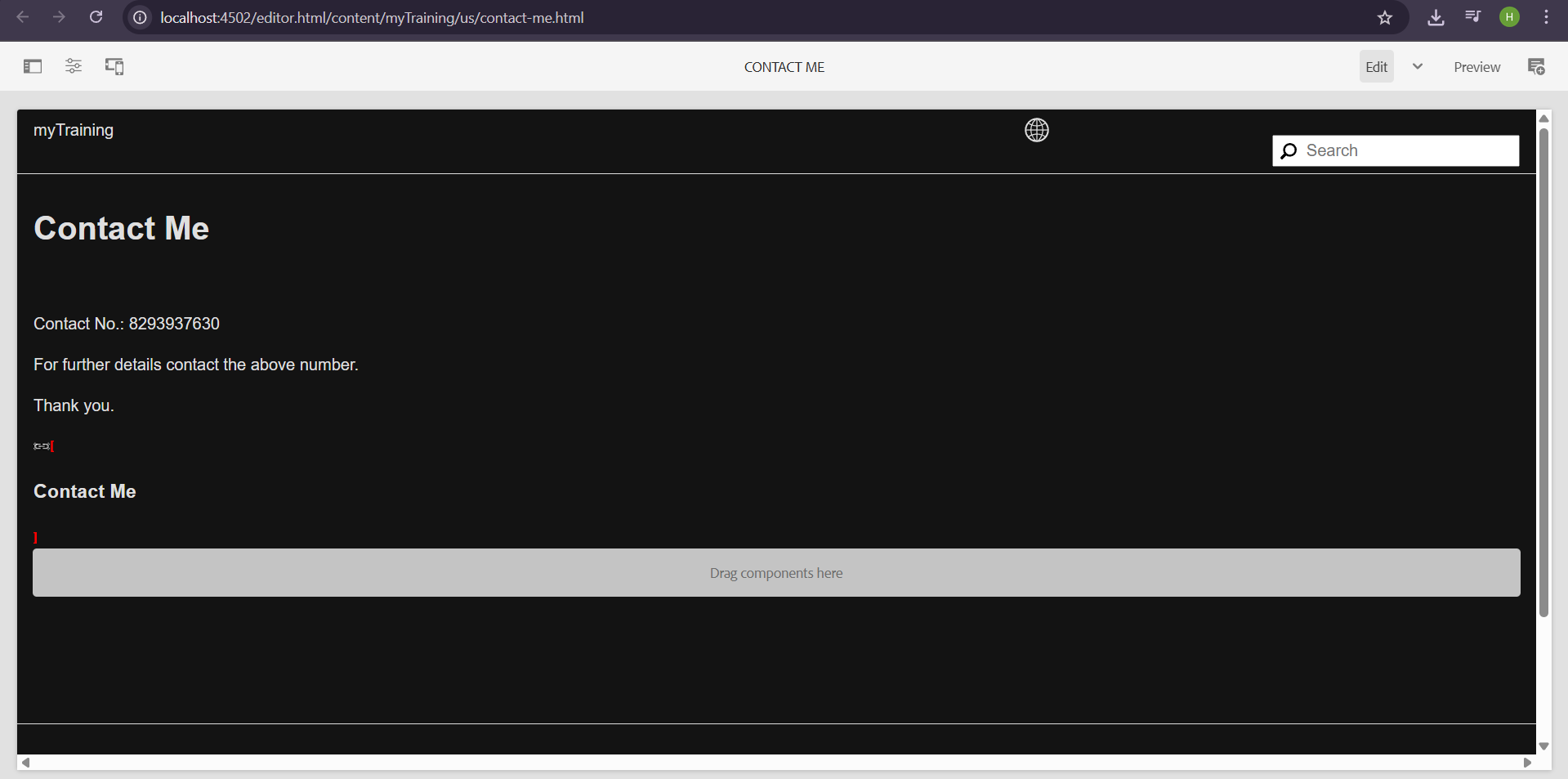
- News (Menu item linking to the News section)  
 - Contact Us Page  
 - About Me Page



## 3. Create Contact Us Page

• Use Text Component to add:

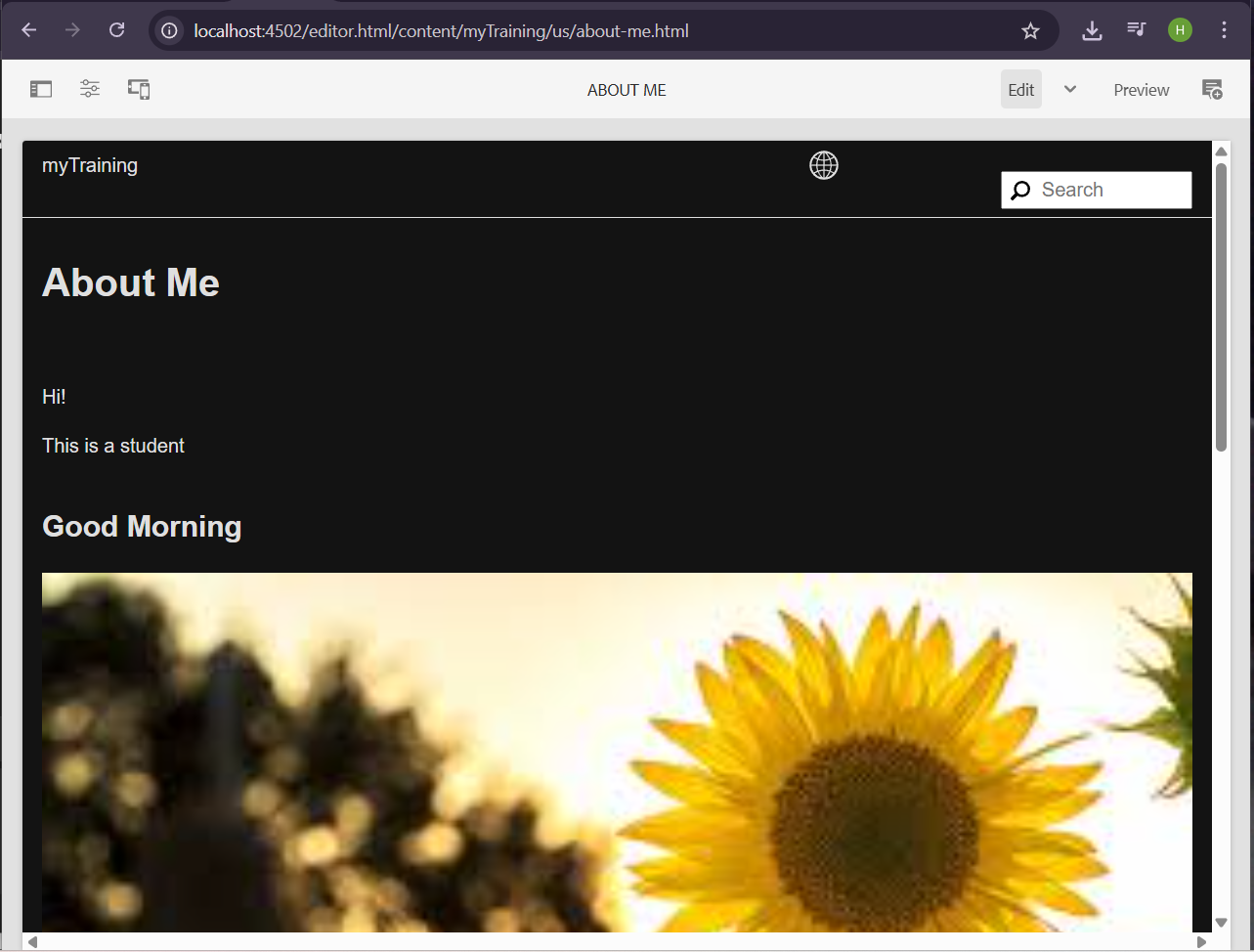
- Mobile Number  
 - Office Address  
 - Email Address



## 4. Create About Me Page

• Use Teaser or Image Component to display a journalist image.

• Use Text Component for a biography or description.





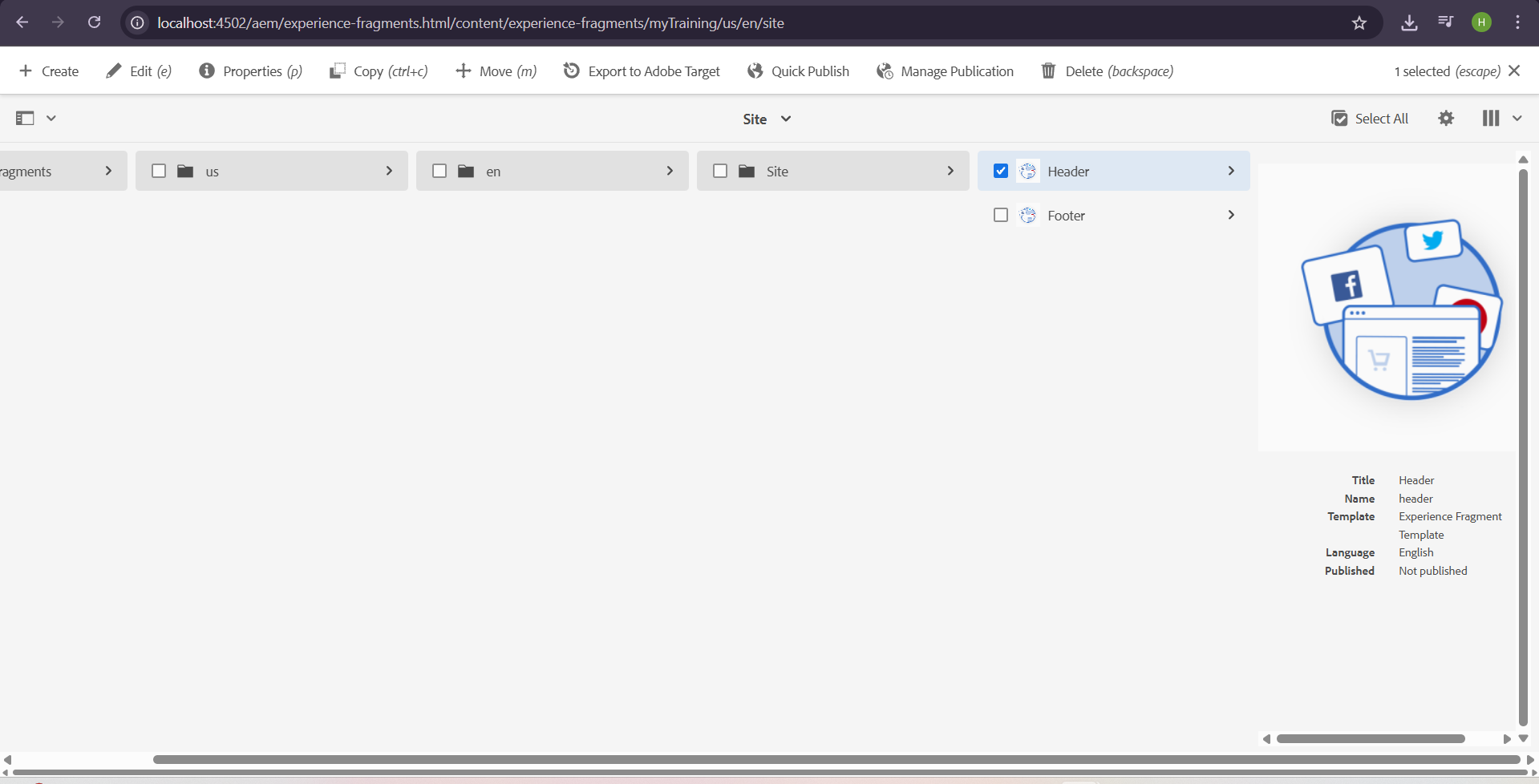
## 5. Create Footer XF with 4 Sections

• News Menu Section: Use List Component to display 4 recent news articles.

• About Me Section: Use Text Component to provide content about the journalist.

• Contact Us Section: Use Text Component for contact details.

• Social Media Section: Use List Component to provide links to social media accounts.



## 6. Create Custom Service to Print "Hello World"

Service Implementation:

@Component(service = HelloWorldService.class, immediate = true)  
public class HelloWorldService {  
 public String getMessage() {  
 return "Hello World";  
 }  
}

Call Service in Sling Model and Print in Logs:

@Model(adaptables = Resource.class, defaultInjectionStrategy = DefaultInjectionStrategy.OPTIONAL)  
public class NewsModel {  
 @OSGiService  
 private HelloWorldService helloWorldService;  
   
 @PostConstruct  
 protected void init() {  
 if (helloWorldService != null) {  
 String message = helloWorldService.getMessage();  
 LOGGER.info("Service Message: {}", message);  
 }  
 }  
}

## 7. Create Custom Configurations for 3rd Party API

OSGi Configuration Interface:

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

Service Implementation:  
@Component(service = APIService.class, immediate = true)  
@Designate(ocd = APIConfig.class)  
public class APIService {  
 private String apiUrl;  
  
 @Activate  
 @Modified  
 protected void activate(APIConfig config) {  
 this.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) {  
 LOGGER.error("Error fetching API data", e);  
 }  
 return "";  
 }  
}

Fetch and Print API Data in Logs:  
@Model(adaptables = Resource.class, defaultInjectionStrategy = DefaultInjectionStrategy.OPTIONAL)  
public class APIModel {  
 @OSGiService  
 private APIService apiService;  
  
 @PostConstruct  
 protected void init() {  
 if (apiService != null) {  
 String jsonData = apiService.fetchAPIData();  
 LOGGER.info("API Response: {}", jsonData);  
 }  
 }  
}

## 

## Deployment & Configuration Steps:

1. Create News Pages under /content/us/en/news and use News Component.

2. Create Header XF with a menu linking to News, About Me, and Contact Us.

3. Create Contact Us and About Me pages using text, teaser, and image components.

4. Create Footer XF with News List, About Me, Contact, and Social Media.

5. Deploy Custom Service to print "Hello World" and call it in Sling Model.

6. Configure API OSGi Service and fetch JSON data from a third-party API.

7. Test Logging for service calls and API fetch.