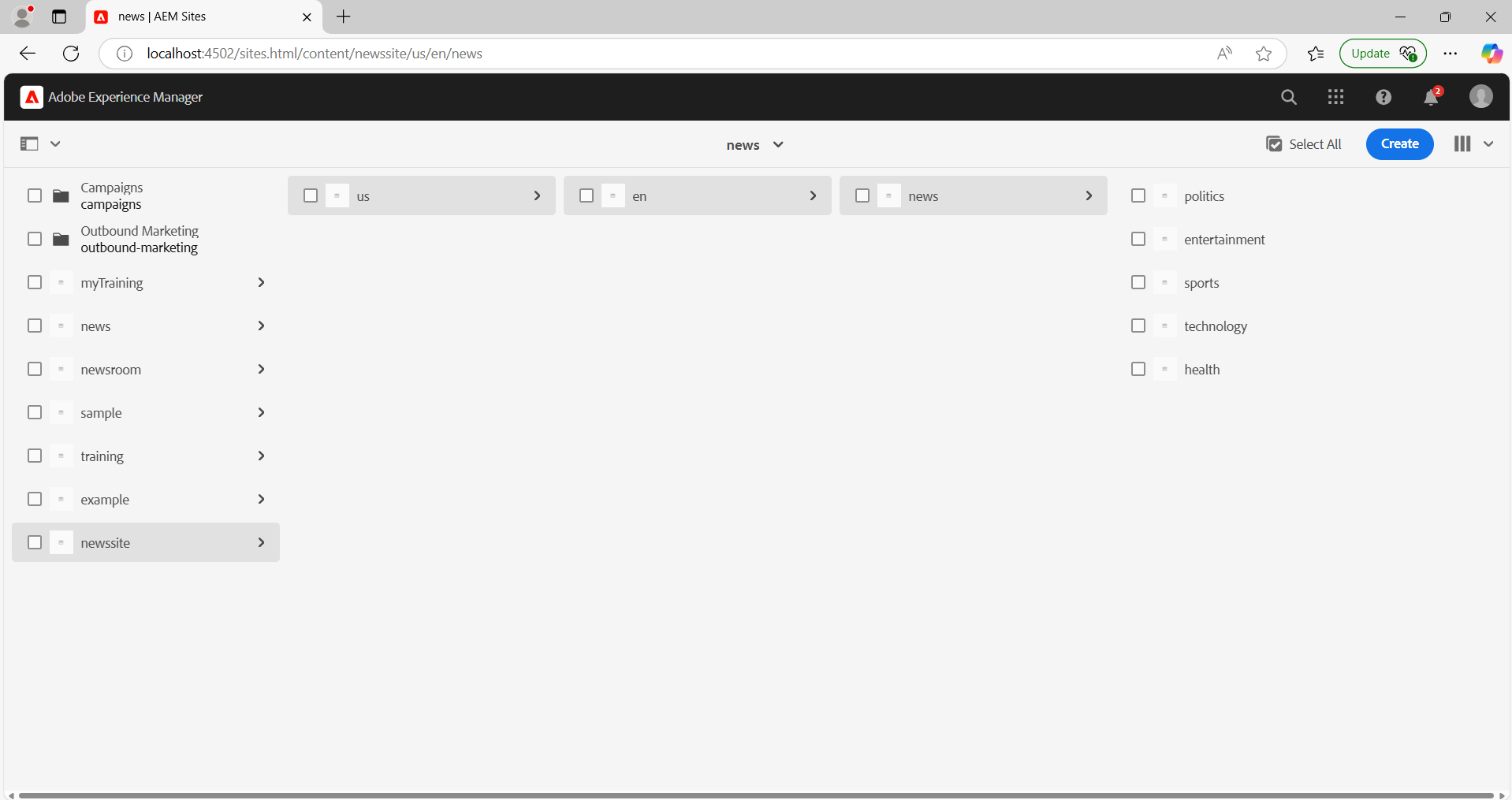
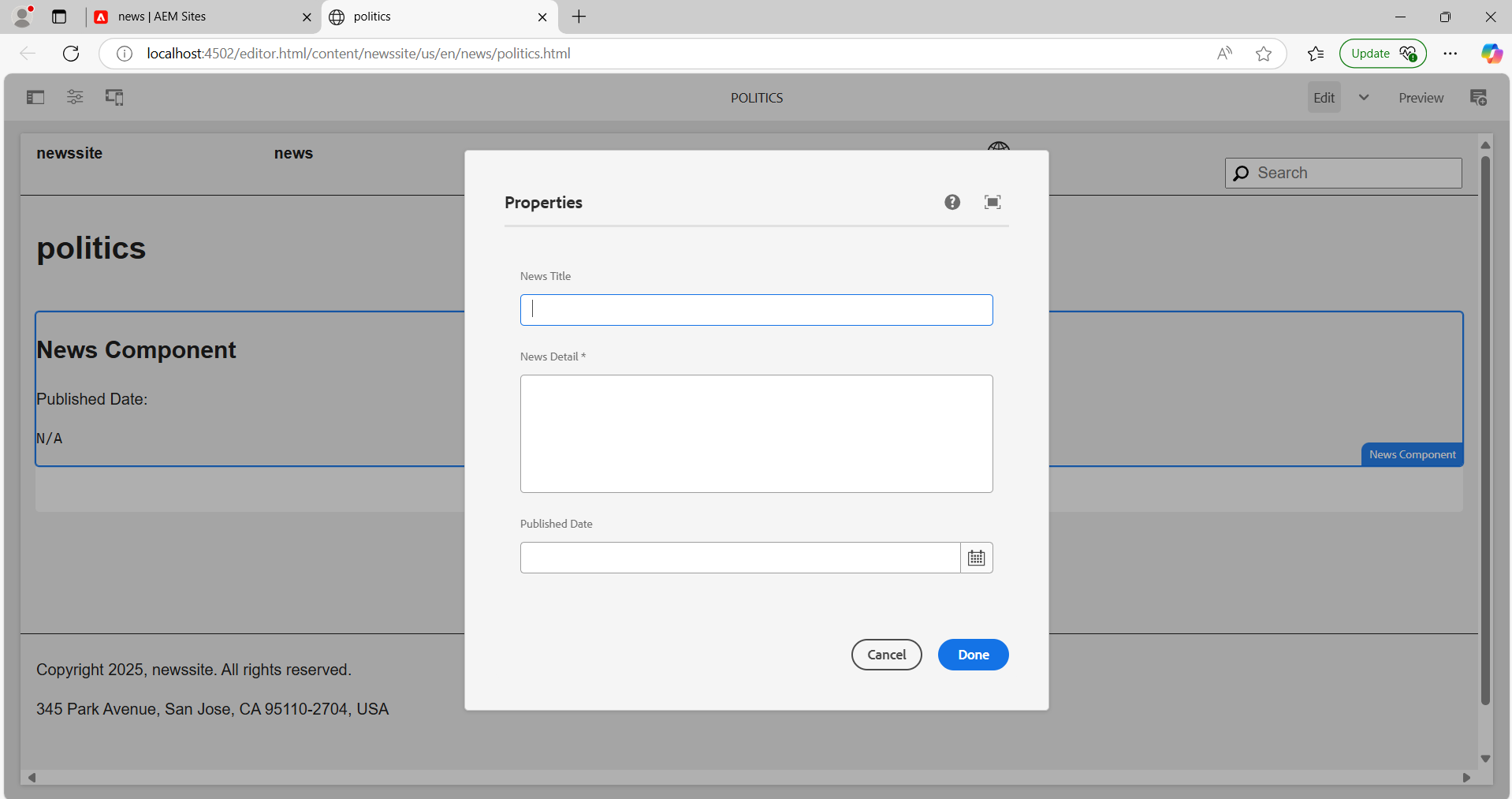
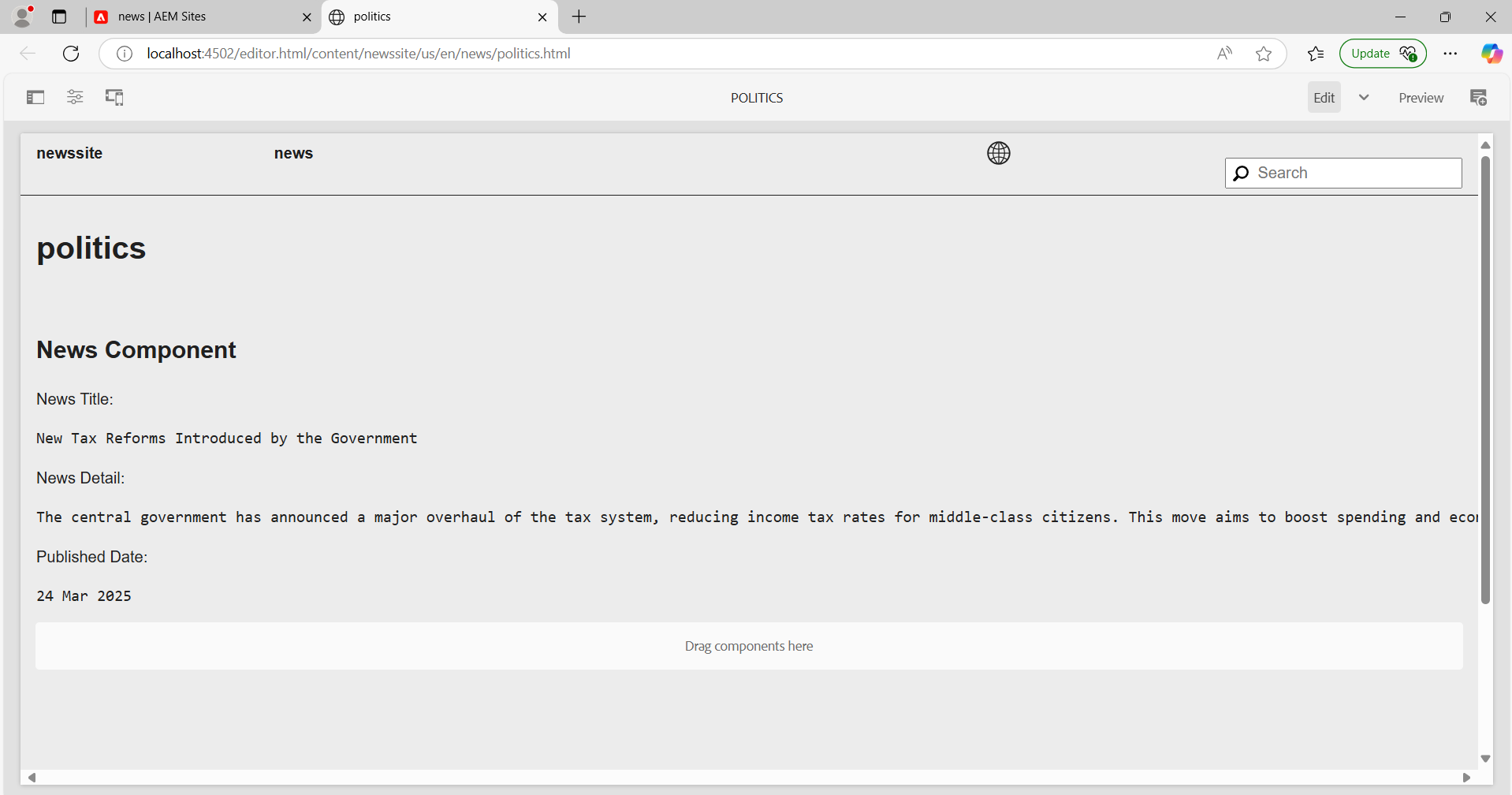
**1. Create 5 news article pages under /content/us/en/news**

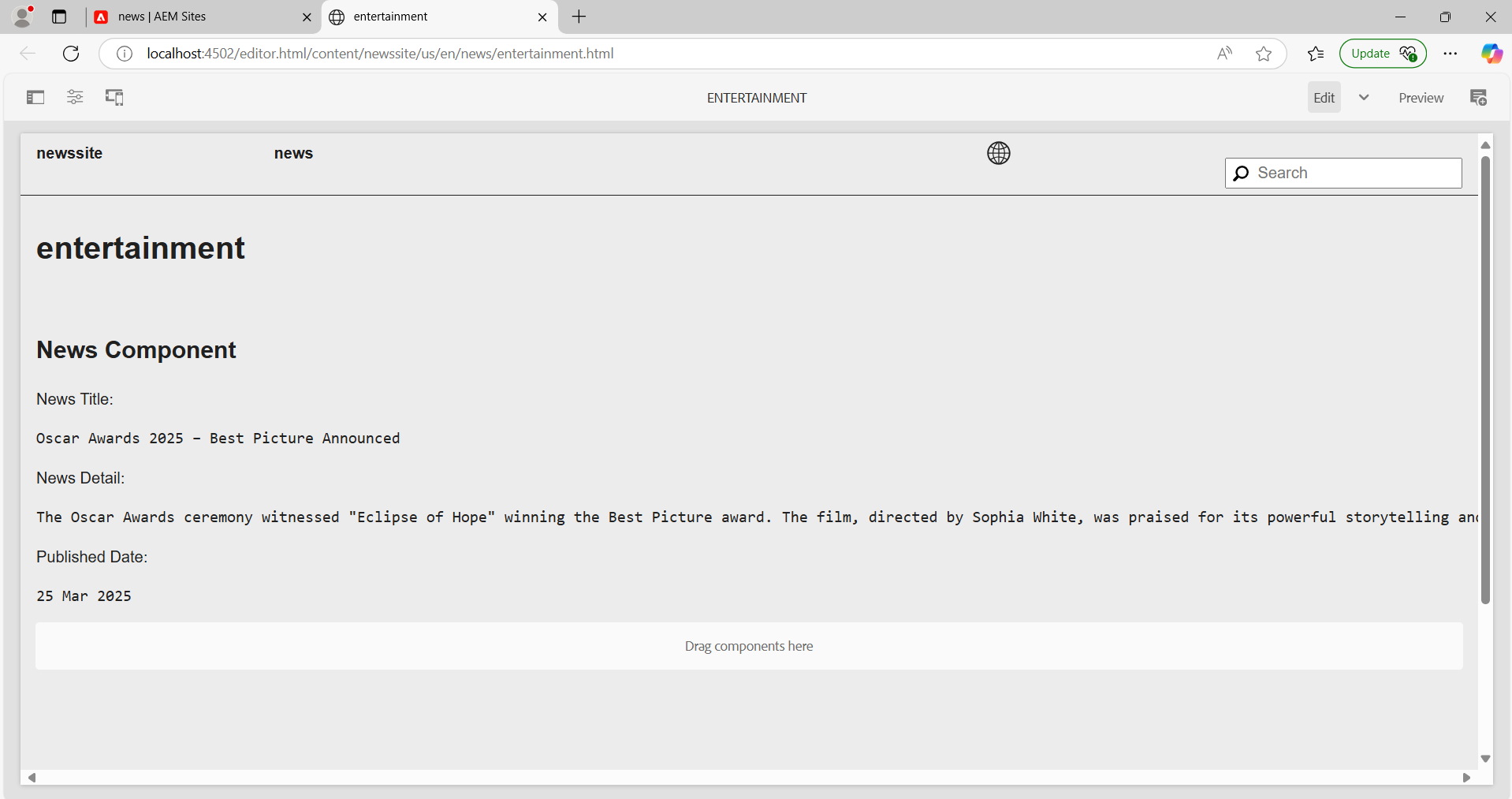
**All should be unique.**

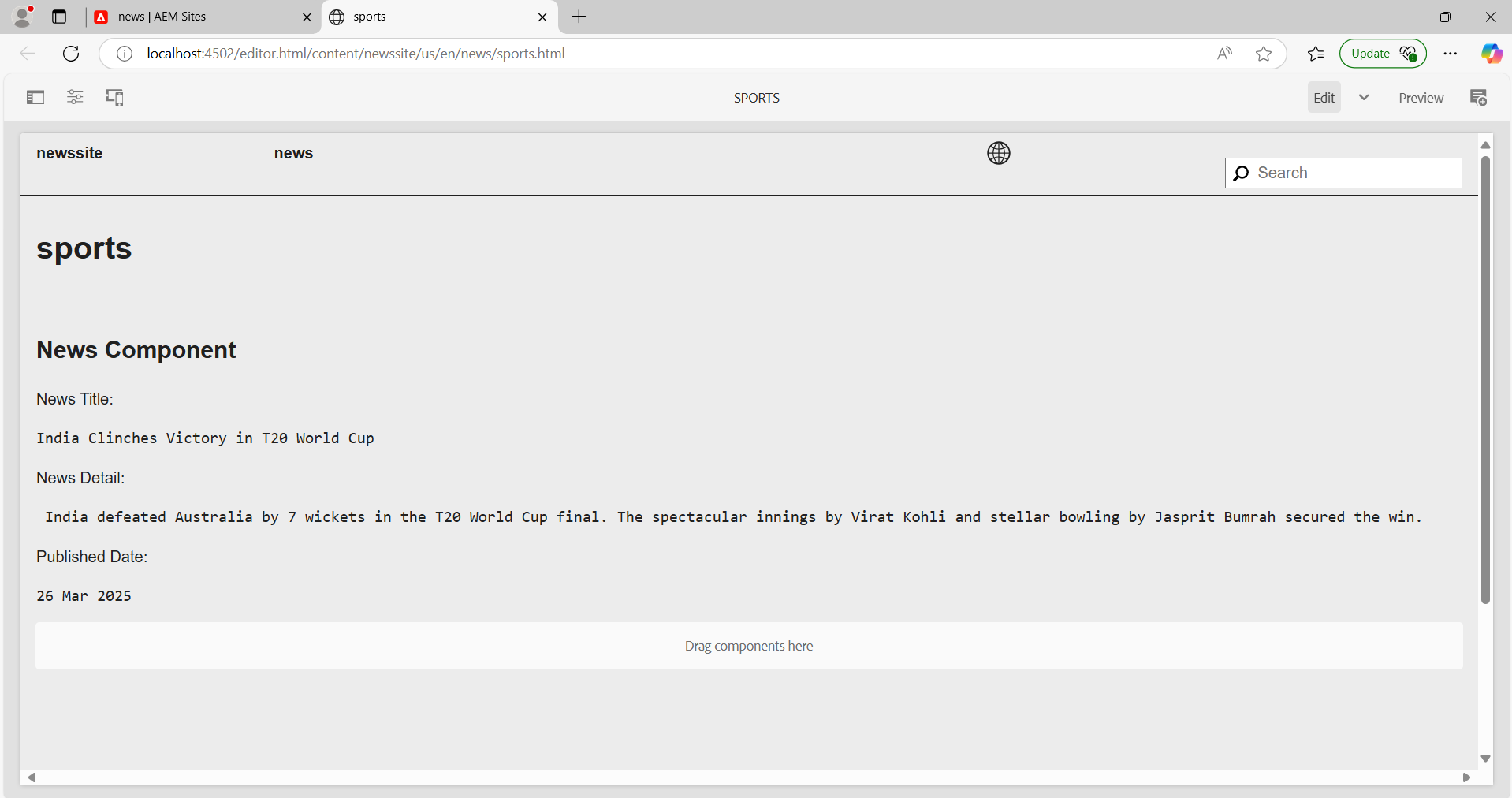
****

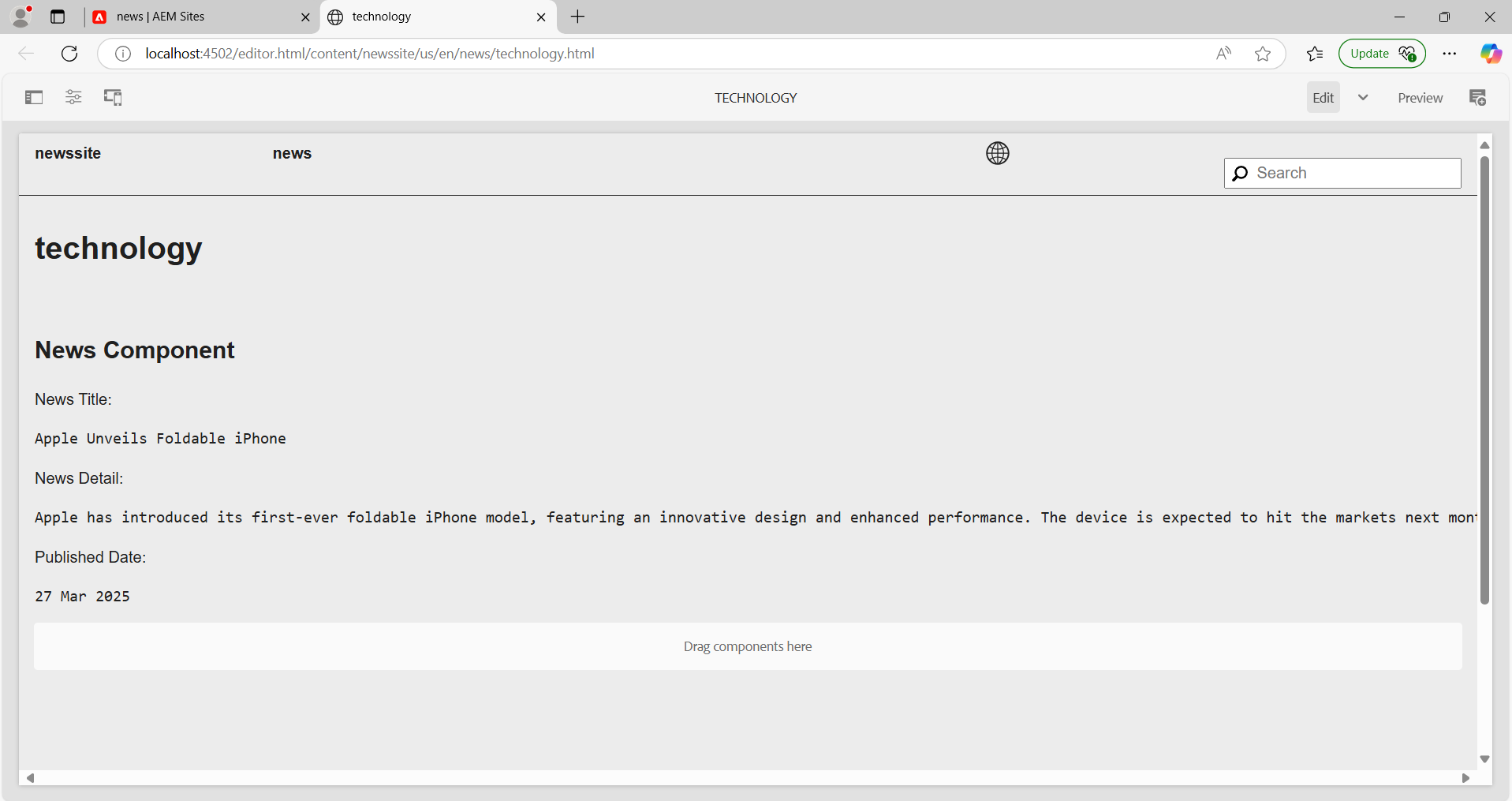
**2. Use the News component which we have created previously to provide the title, news details and published date.**

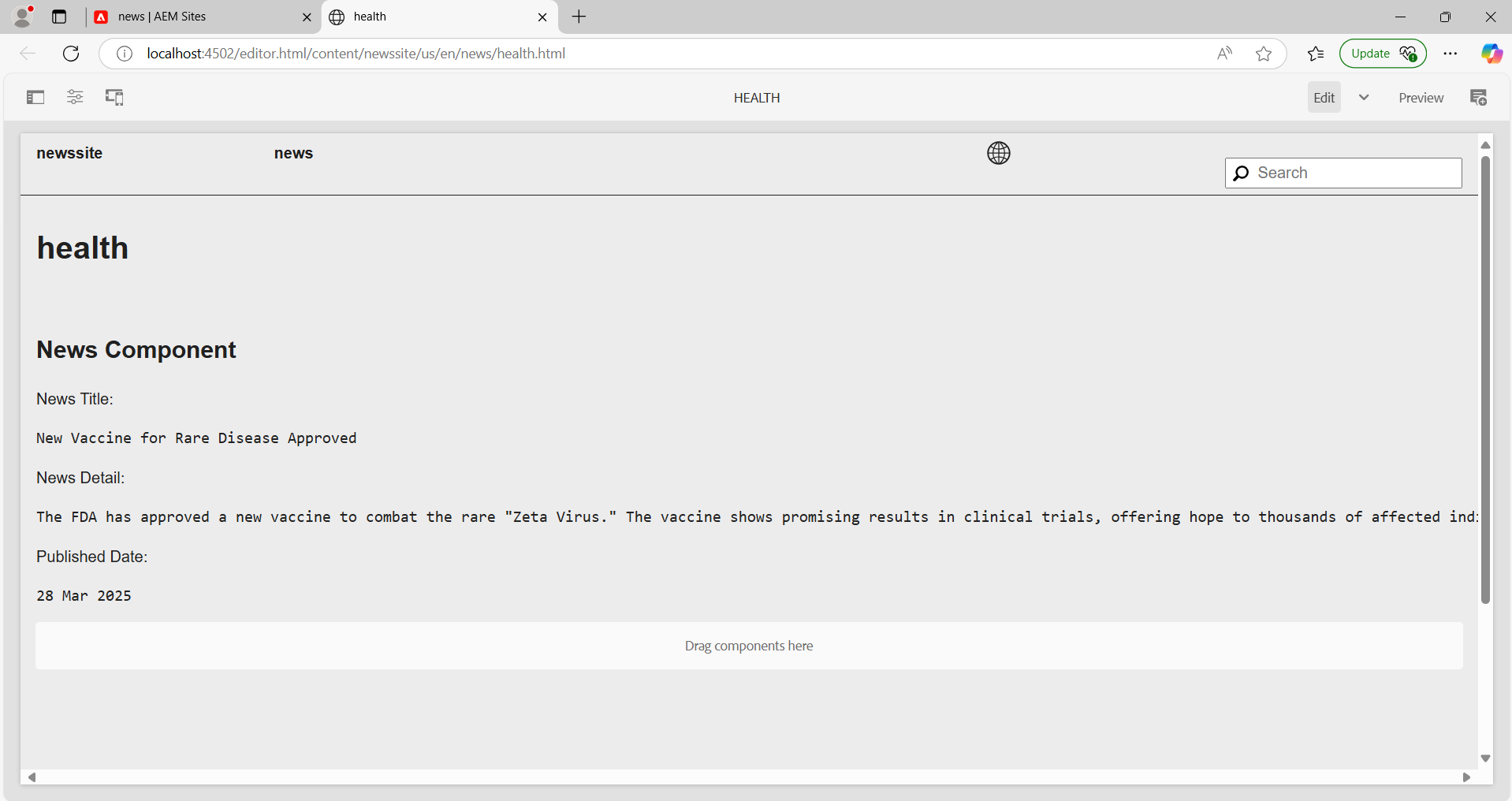
****

****

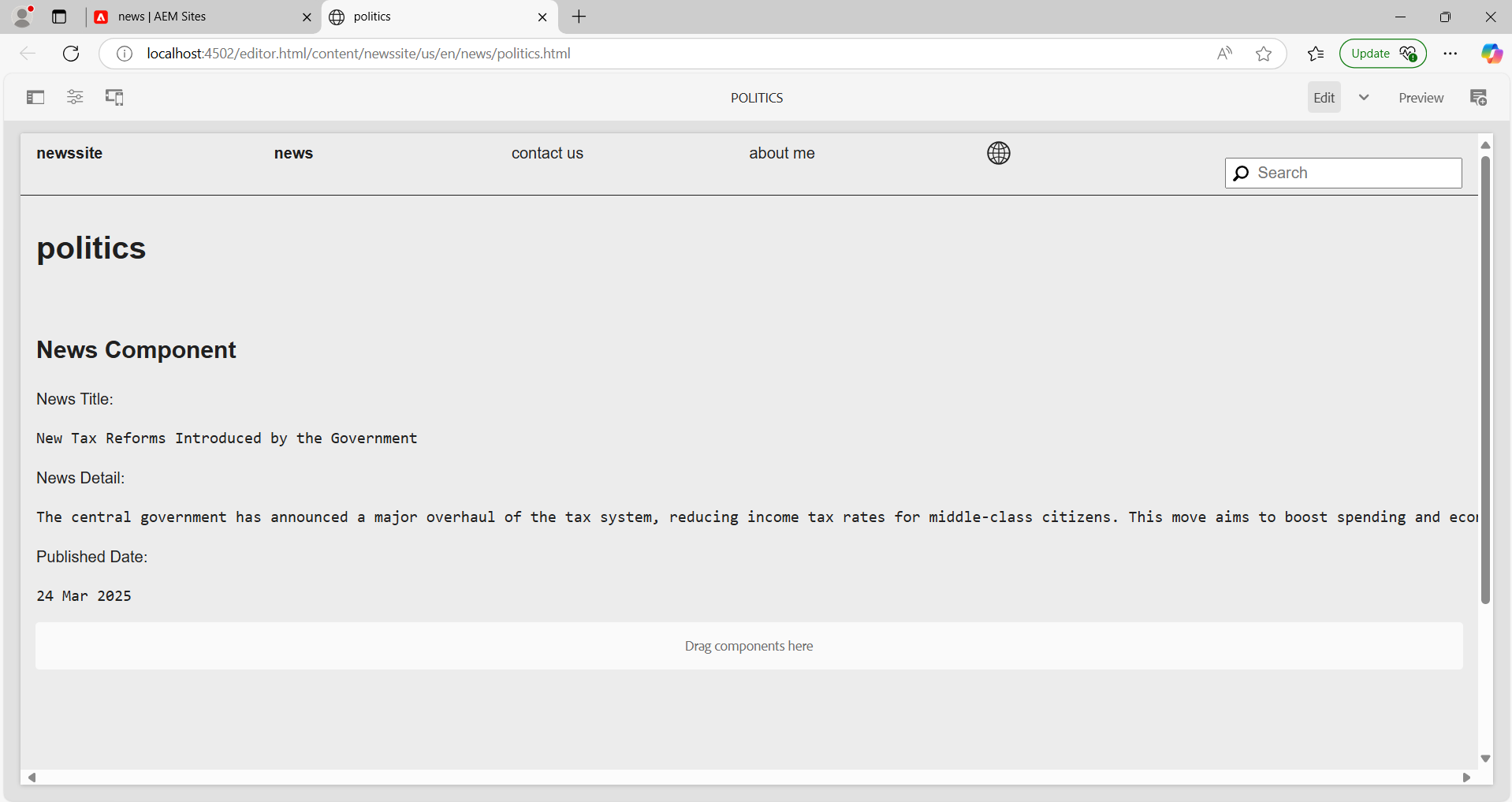
****

****

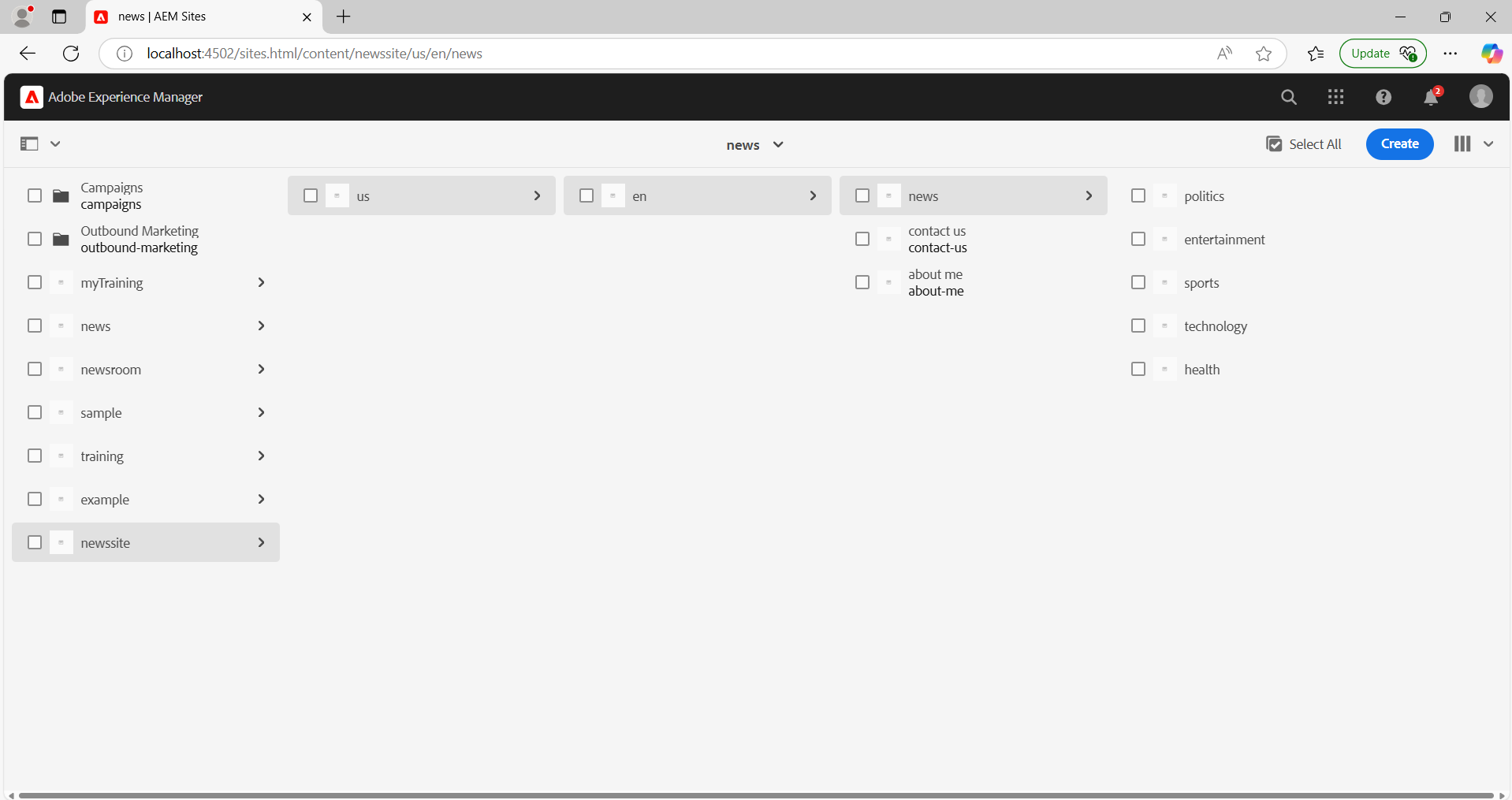
****

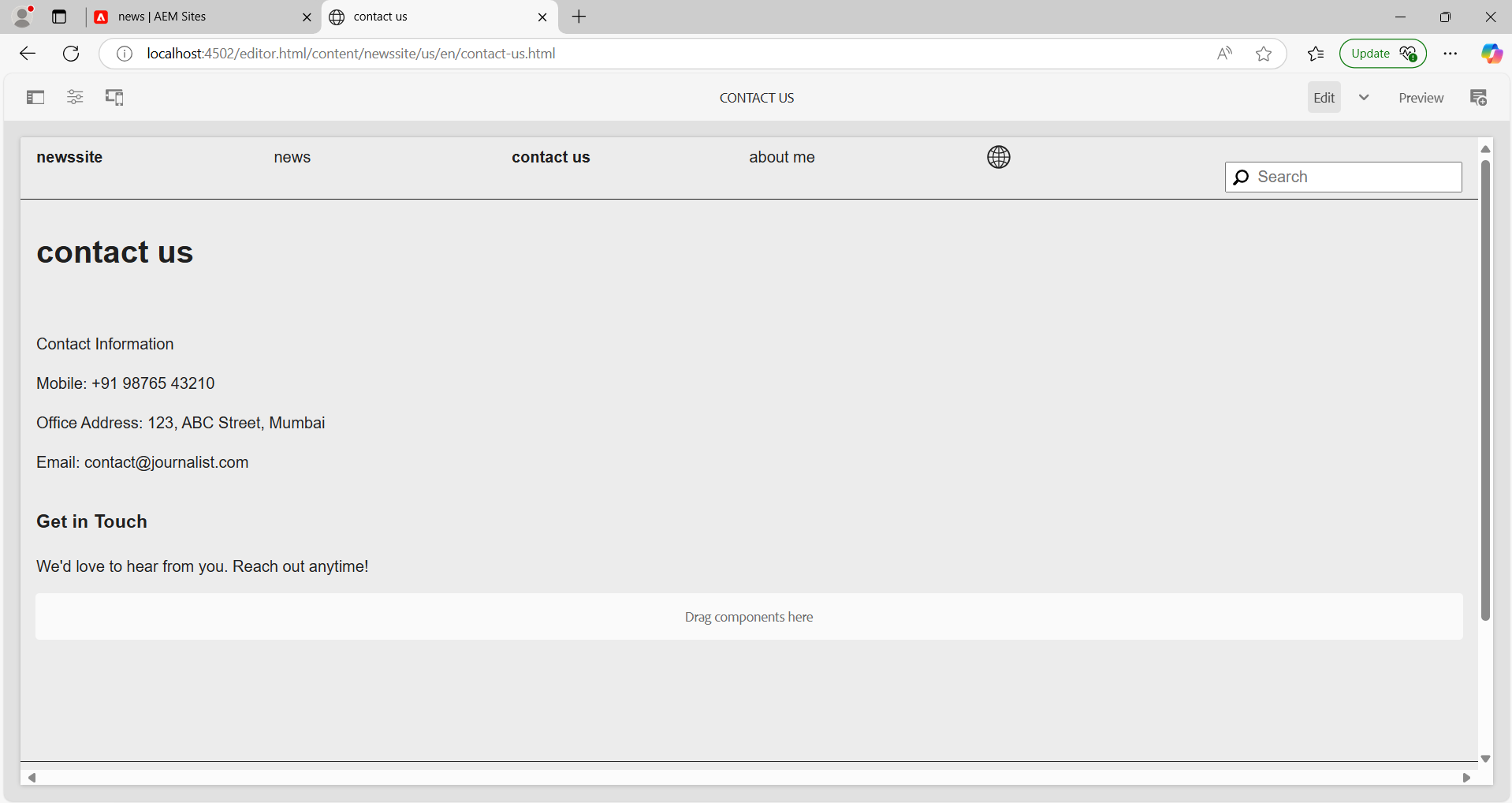
****

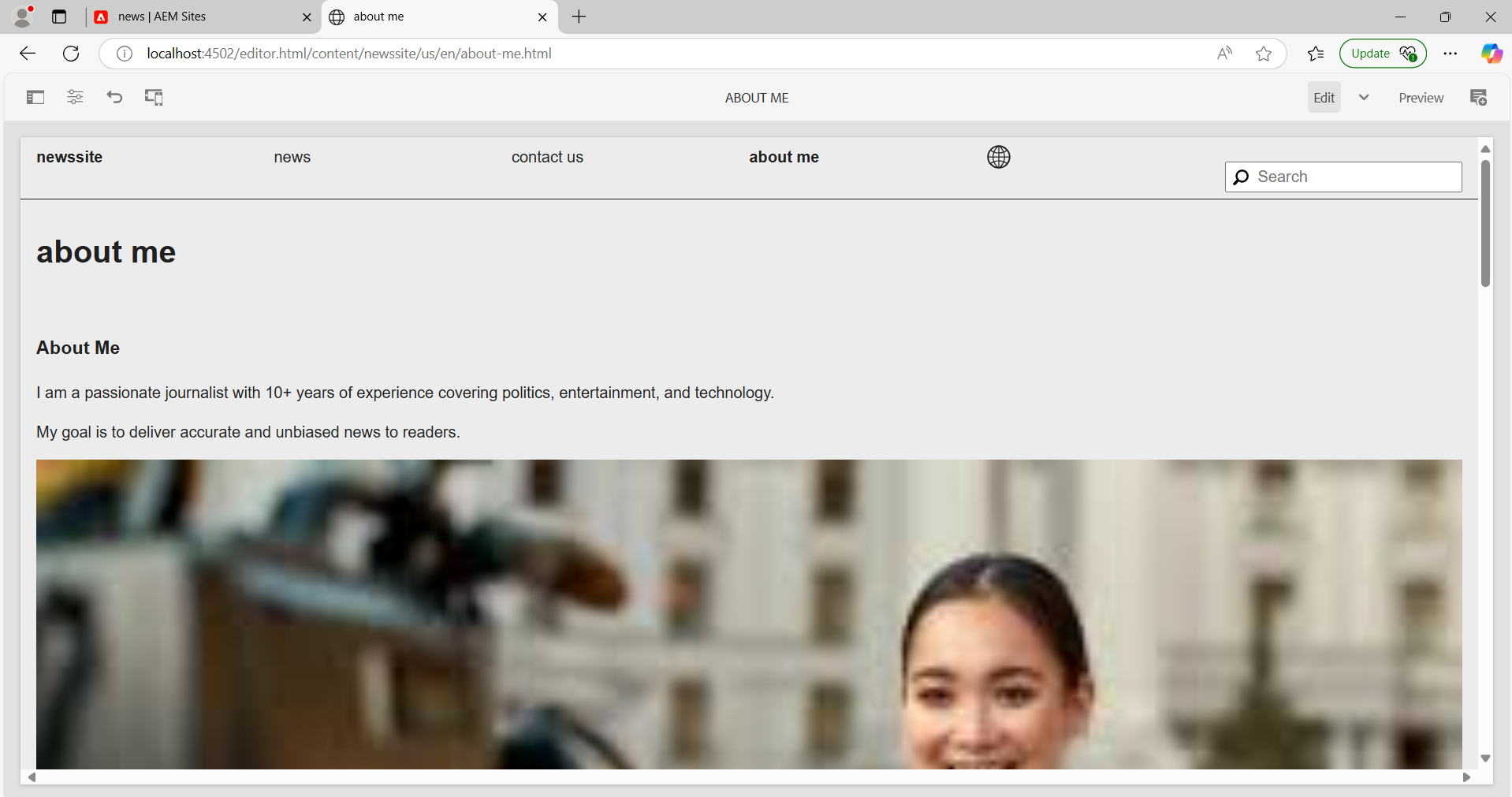
**3. Create a Header Experience fragment for the header and use this page as a menu (name should be news).**

****

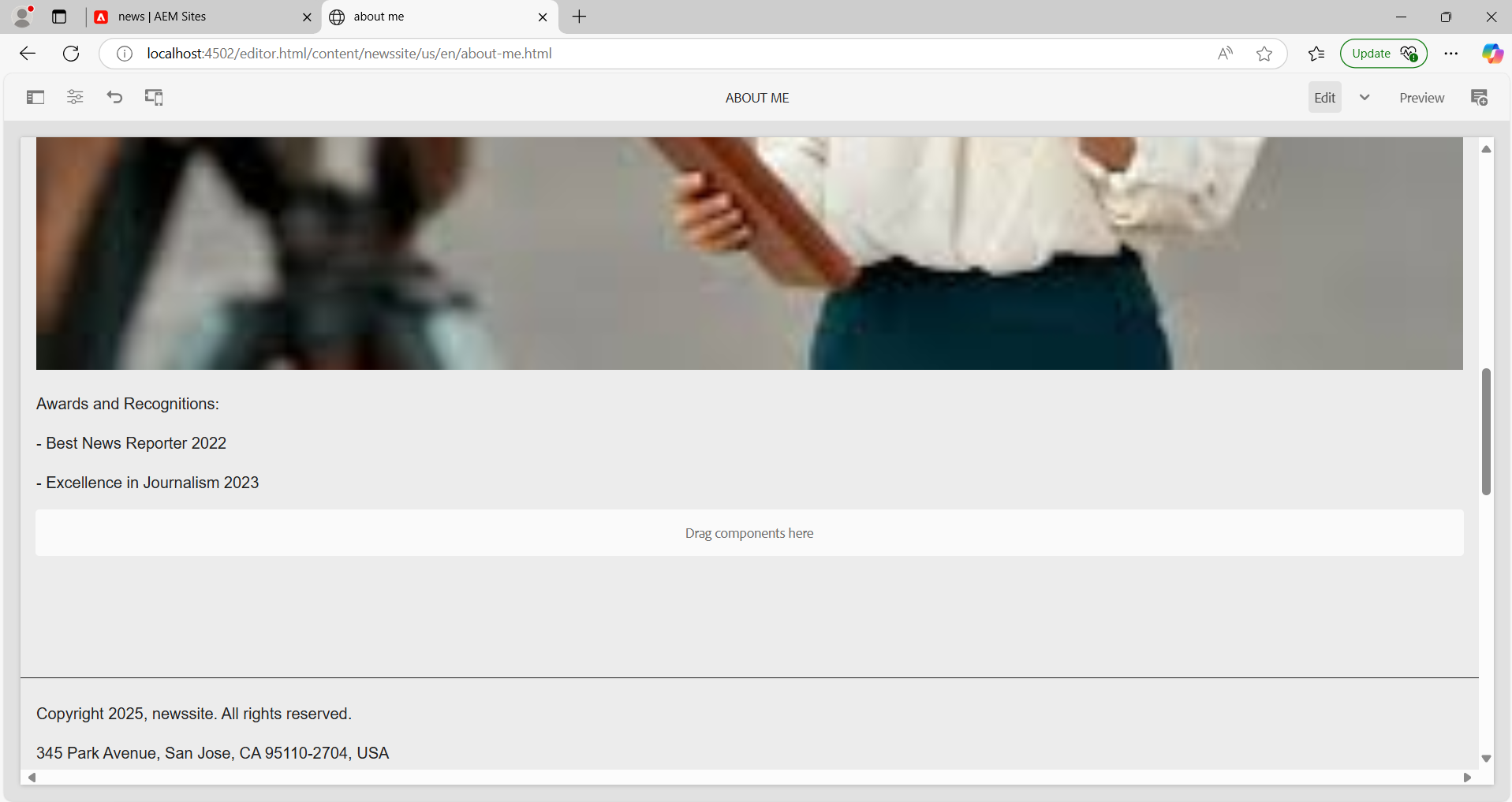
**4. Create the Contact Us page and About me page (you can use teaser or image, text, title components to provide some details about journalists on the about me page and on the contact us page there should be some contact related details it can have their mobile number office address or email address).**

****

****

****

****

****

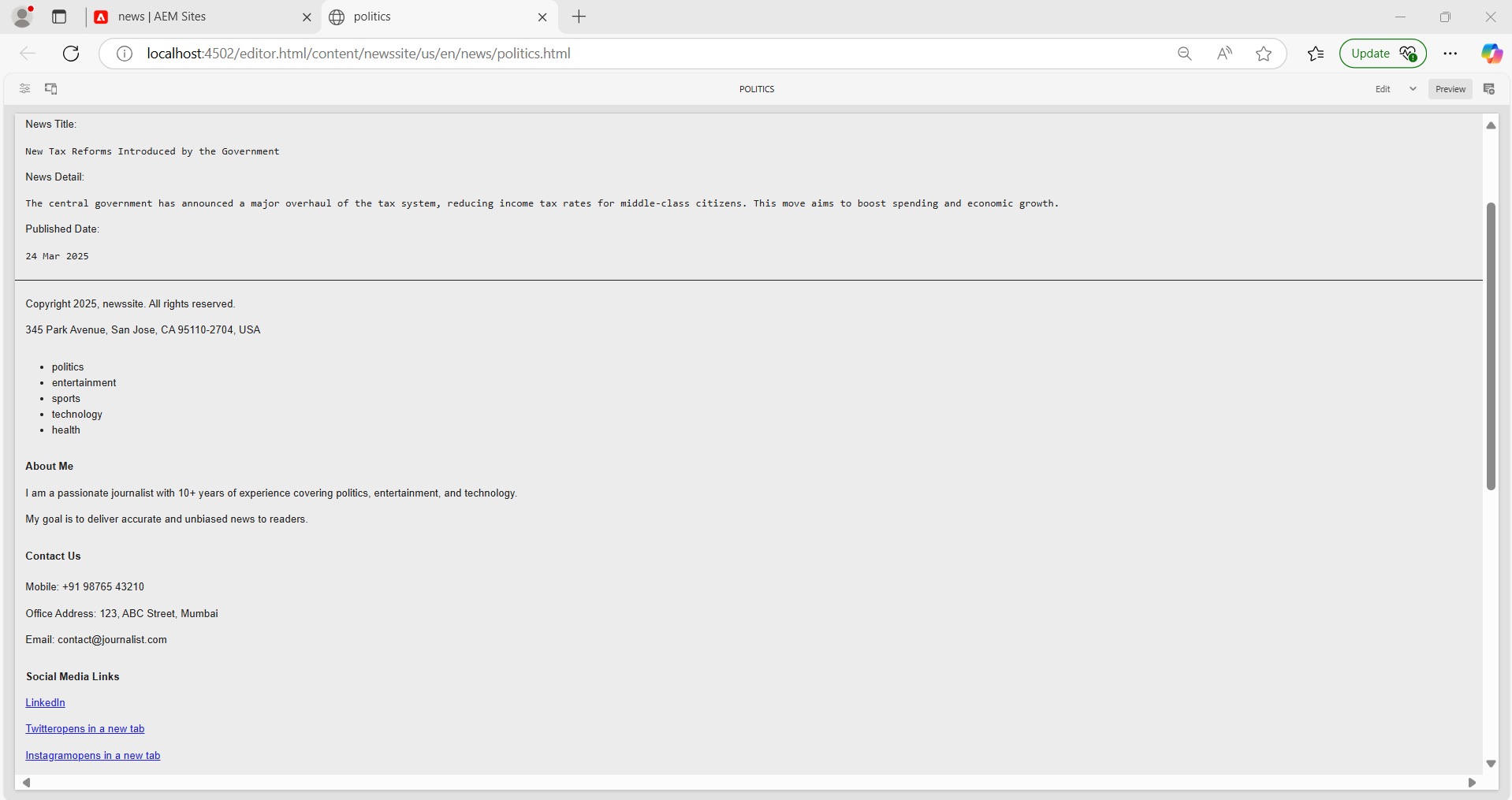
**5. Create footer XF and it could have 4 sections:**

**1. The news menu section can have 4 news articles. You can use the list component for this.**

**2. The About me section can have some content and you can use text components.**

**3. Contact us section-you can have some content, and you can use text components**

**4. Social media section-you can use the list component for this to provide some social media account references.**

****

**6. Create a custom service to print Hello World and call this service from the news component sling model and print this value in logs as well.**

Create a service class:  
Path: /apps/newssite/core/src/main/java/com/newssite/core/services

**HelloWorldService.java**

package com.newssite.core.services;

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

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

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@Component(service = HelloWorldService.class)

public class HelloWorldService {

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

public String getMessage() {

String message = "Hello World from Newsroom Service";

LOG.info(message);

return message;

}

}

Update the NewsModel.java to call the service:

package com.newssite.core.models;

import com.newssite.core.services.HelloWorldService;

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

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

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

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@Model(adaptables = Resource.class)

public class NewsModel {

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

@OSGiService

private HelloWorldService helloWorldService;

public String getServiceMessage() {

String message = helloWorldService.getMessage();

LOG.info("Service message: {}", message);

return message;

}

}

**7. Create custom configurations where I can provide the 3rd party API. For example(https://jsonplaceholder.typicode.com/posts) and I can see the data as json. And this data should be printed in logs.**

Create the configuration:  
Path: /apps/newssite/core/src/main/java/com/newssite/core/config

**NewsAPIConfig.java**

package com.newssite.core.config;

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

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

@ObjectClassDefinition(name = "Newsroom API Configuration")

public @interface NewsAPIConfig {

@AttributeDefinition(name = "API Endpoint")

String apiEndpoint() default "https://jsonplaceholder.typicode.com/posts";

}

Create the service to call the API:  
Path: /apps/newssite/core/src/main/java/com/newssite/core/services

**NewsAPIService.java**

package com.newssite.core.services;

import com.newssite.core.config.NewsAPIConfig;

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

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

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

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

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

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.apache.http.client.methods.CloseableHttpResponse;

import org.apache.http.impl.client.CloseableHttpClient;

import org.apache.http.impl.client.HttpClients;

import org.apache.http.client.methods.HttpGet;

import org.apache.http.util.EntityUtils;

@Component(service = NewsAPIService.class, configurationPolicy = ConfigurationPolicy.REQUIRE)

public class NewsAPIService {

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

@Reference

private NewsAPIConfig config;

@Activate

@Modified

protected void activate(NewsAPIConfig config) {

this.config = config;

}

public void fetchAPIData() {

try (CloseableHttpClient client = HttpClients.createDefault()) {

HttpGet request = new HttpGet(config.apiEndpoint());

CloseableHttpResponse response = client.execute(request);

String jsonResponse = EntityUtils.toString(response.getEntity());

LOG.info("API Response: {}", jsonResponse);

} catch (Exception e) {

LOG.error("Failed to fetch API data", e);

}

}

}

Add the configuration in the OSGi Console:

* Go to http://localhost:4502/system/console/configMgr
* Find "Newssite API Configuration"
* Add the API URL: https://jsonplaceholder.typicode.com/posts
* Save the configuration

Call the service in your component to log the API data.