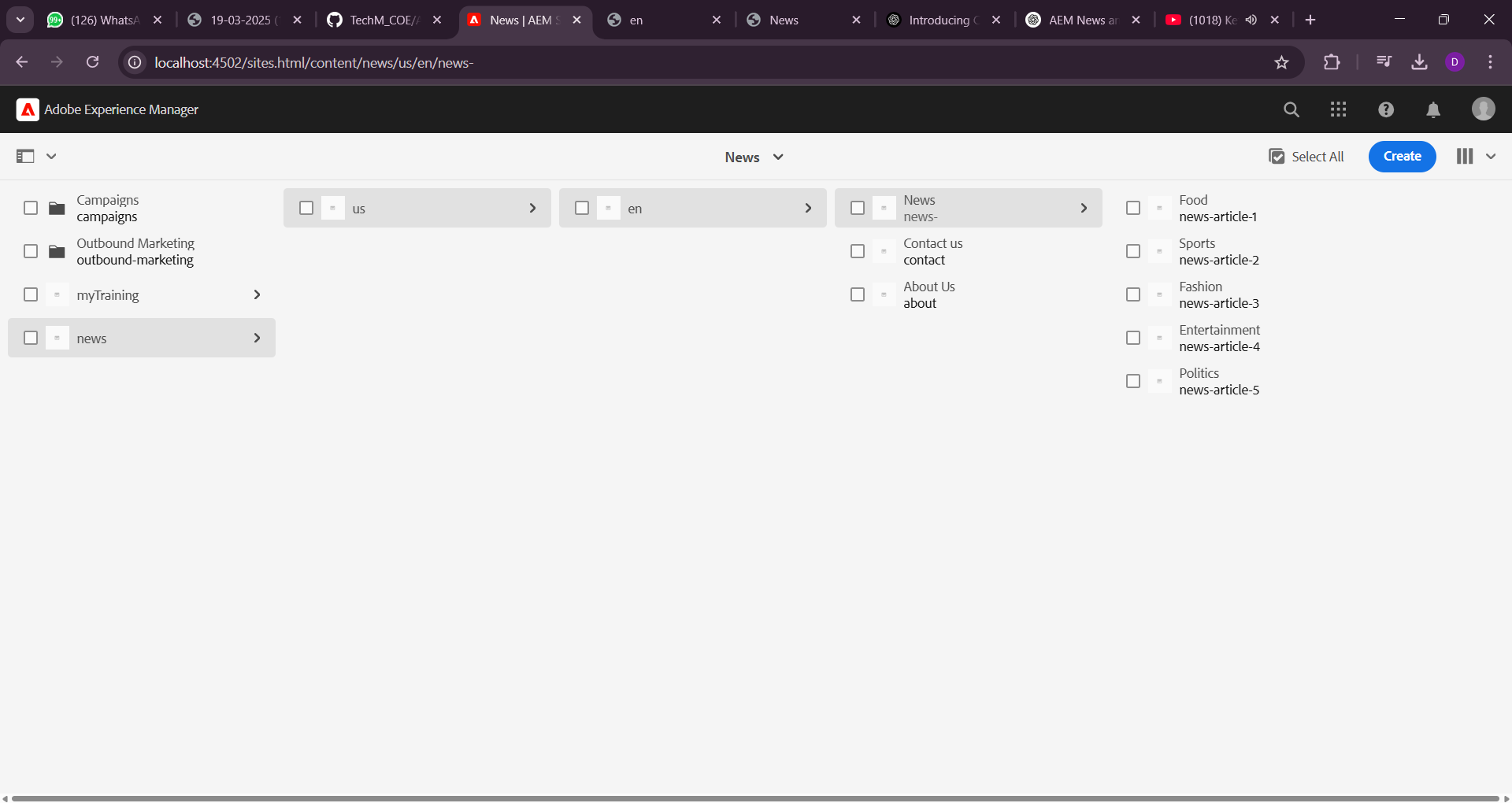
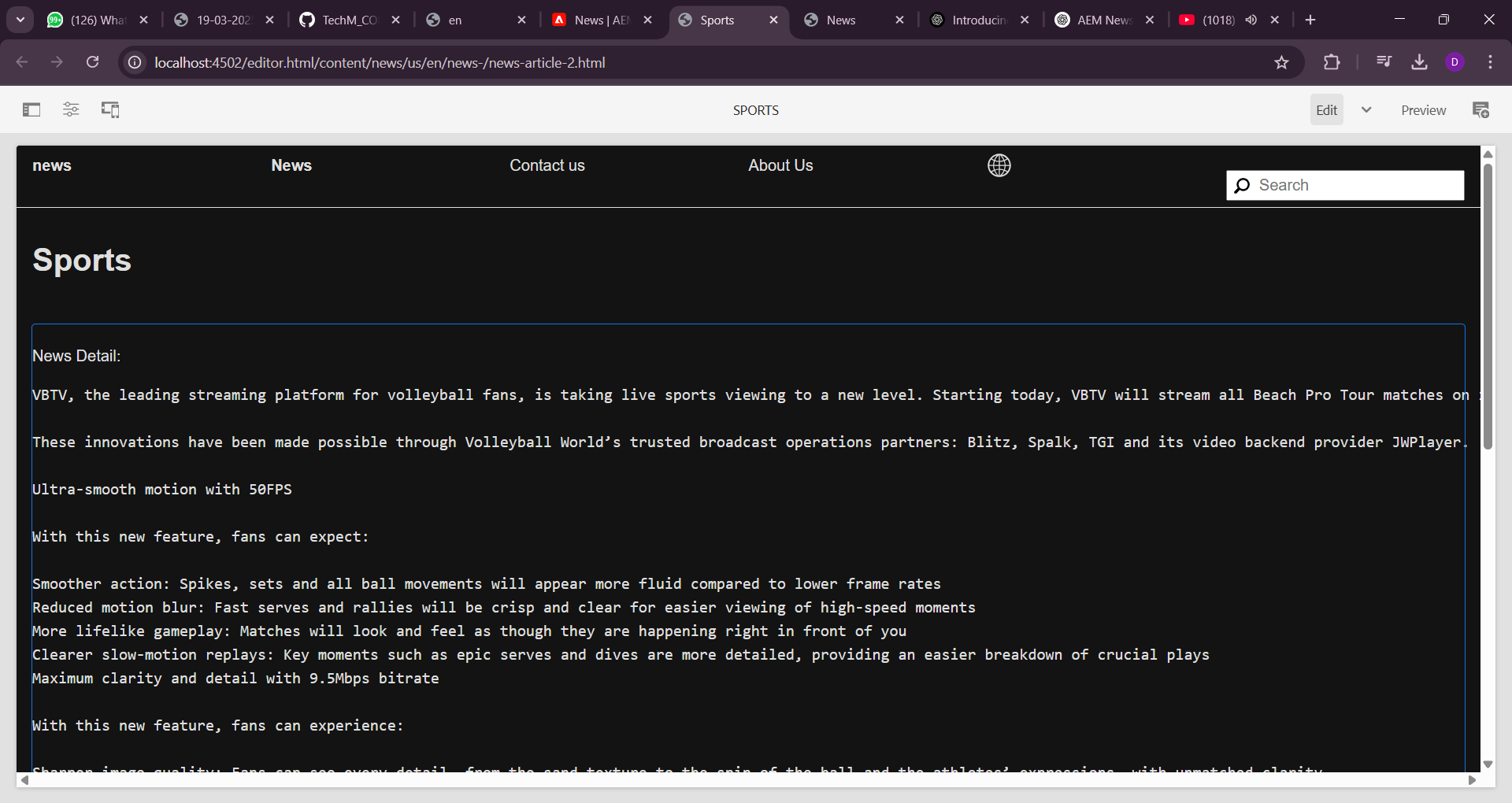
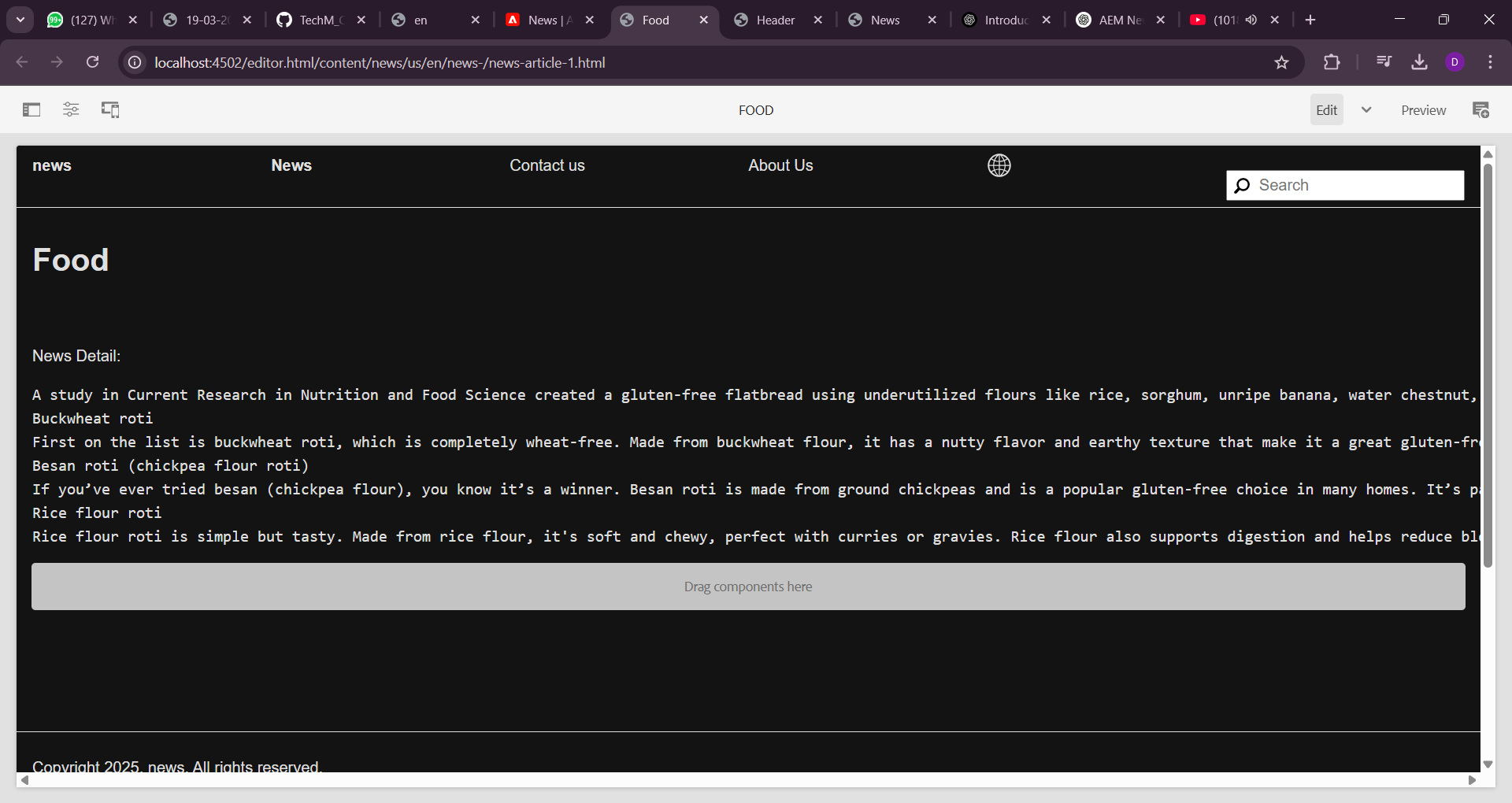
1. **Create 5 News Article Pages** under /content/us/en/news

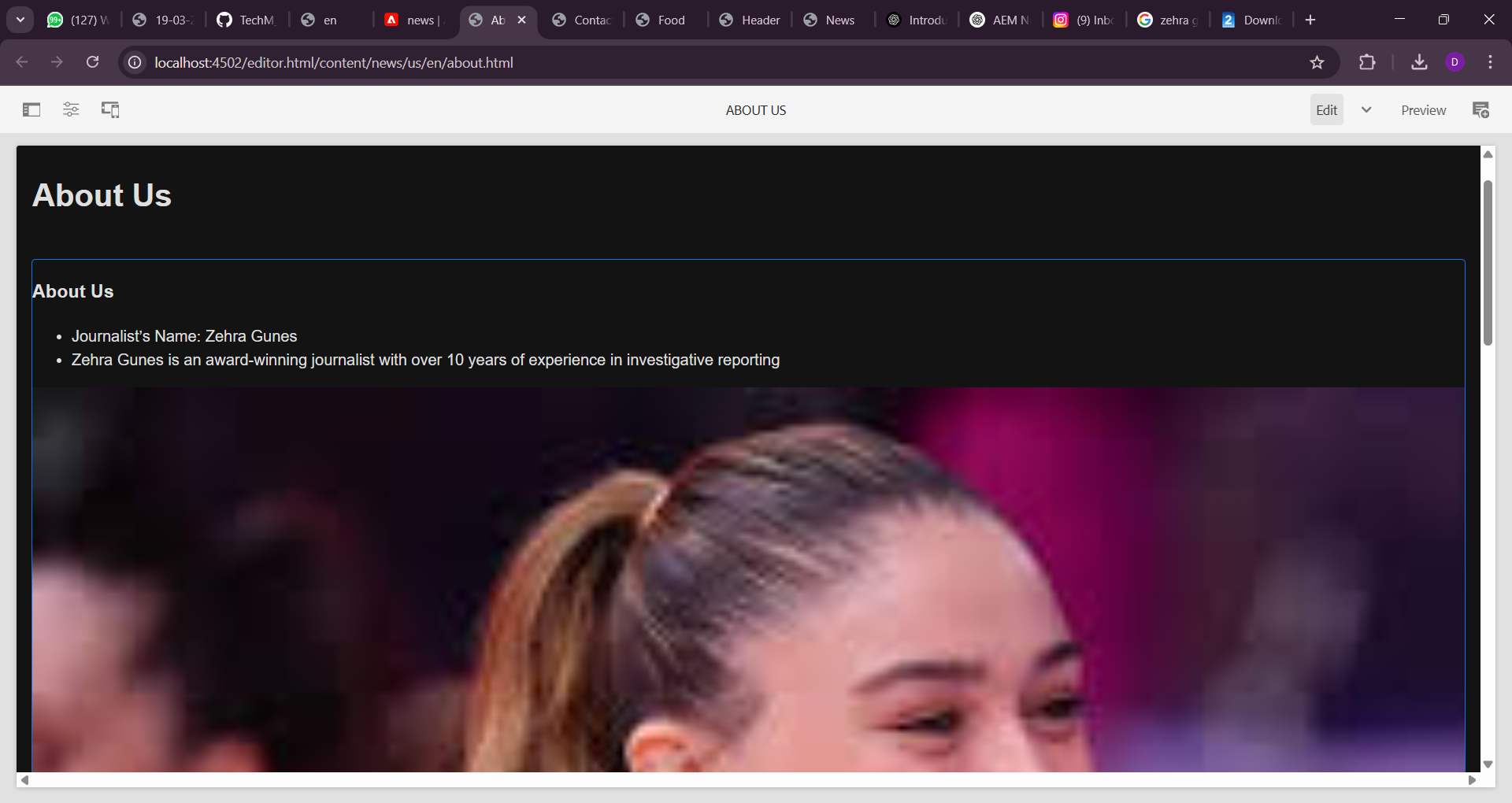




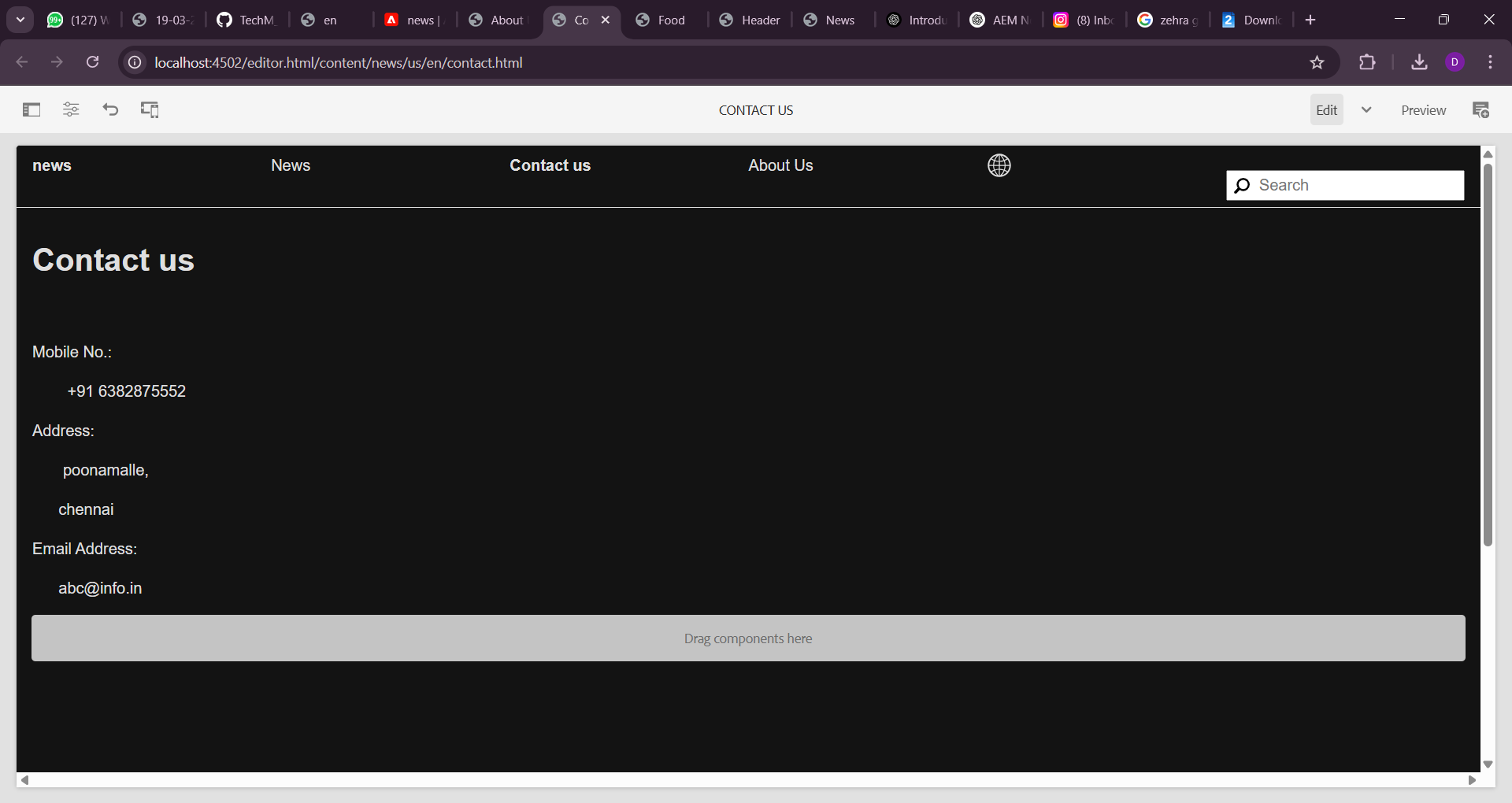
**Create Header Experience fragment**

****

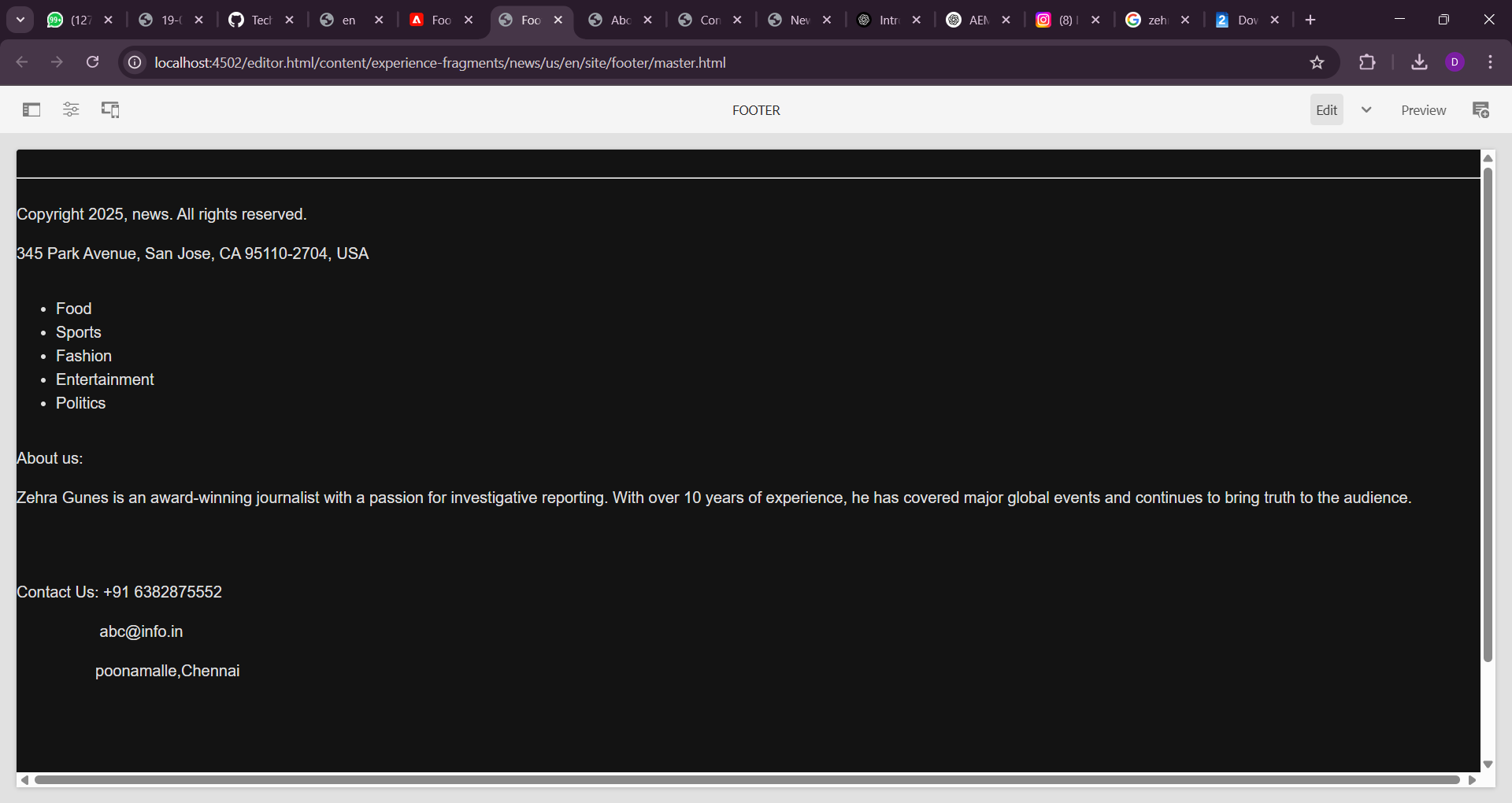
**About Us**

****

**Contact Us**

****

**Footer**

****

**5. Create Custom Service to Print "Hello World"**

**Steps:**

1. **Create a Sling Service:**

Define an OSGi service that returns "Hello World"

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

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

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

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@Component(service = HelloWorldService.class, immediate = true)

public class HelloWorldService {

public String getMessage() {

return "Hello World";

}

}

**Inject Service in Sling Model (News Component Model)**

import javax.annotation.PostConstruct;

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

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

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

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

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

public class NewsModel {

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

@Reference

private HelloWorldService helloWorldService;

@PostConstruct

protected void init() {

if (helloWorldService != null) {

LOG.info("Service Message: {}", helloWorldService.getMessage());

} else {

LOG.error("HelloWorldService is not available.");

}

}

}"Hello World" will be logged in the AEM logs when the news component is used.

**6. Create Custom Configurations for External API**

**Steps:**

1. **Create an OSGi Configuration for API URL**
   * Create a config at:  
     /apps/news /config/com.myTraining.NewsApiConfig.xml

<?xml version="1.0" encoding="UTF-8"?>

<jcr:root xmlns:sling="http://sling.apache.org/jcr/sling/1.0"

xmlns:jcr="http://www.jcp.org/jcr/1.0"

jcr:primaryType="sling:OsgiConfig"

apiUrl="https://jsonplaceholder.typicode.com/posts"/>

1. **Create an OSGi Configurable Service**

@Designate(ocd = NewsApiConfig.class)

@Component(service = NewsApiService.class, immediate = true)

public class NewsApiService {

@ObjectClassDefinition(name = "News API Configuration")

public @interface NewsApiConfig {

@AttributeDefinition(name = "API URL", description = "Provide the API URL")

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

}

@Activate

@Modified

protected void activate(NewsApiConfig config) {

Logger logger = LoggerFactory.getLogger(getClass());

logger.info("Configured API URL: {}", config.apiUrl());

}

}

1. **Log API Response in AEM Logs**
   * Fetch data and log the response:

java

CopyEdit

@Component(service = NewsApiCaller.class, immediate = true)

public class NewsApiCaller {

@Reference

private NewsApiService newsApiService;

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

public void fetchNews() {

try {

String apiUrl = newsApiService.getApiUrl();

HttpGet request = new HttpGet(apiUrl);

CloseableHttpClient httpClient = HttpClients.createDefault();

CloseableHttpResponse response = httpClient.execute(request);

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

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

} catch (Exception e) {

LOG.error("Error fetching API data", e);

}

}

}