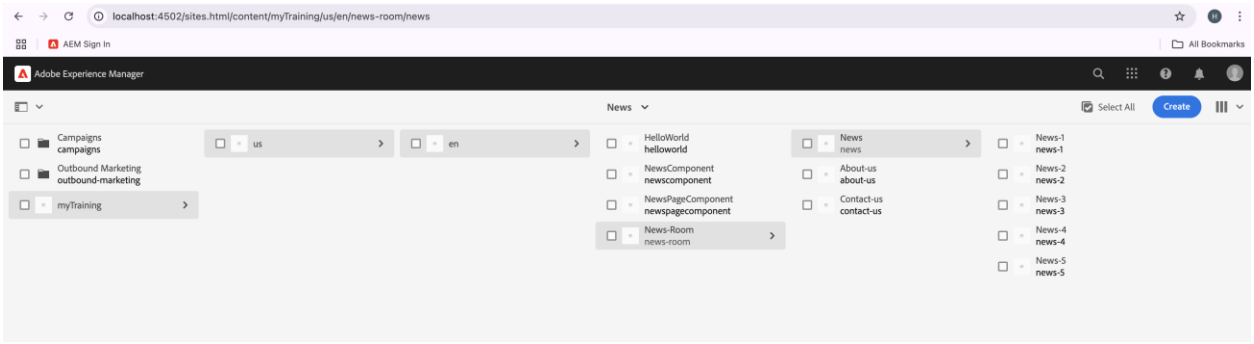
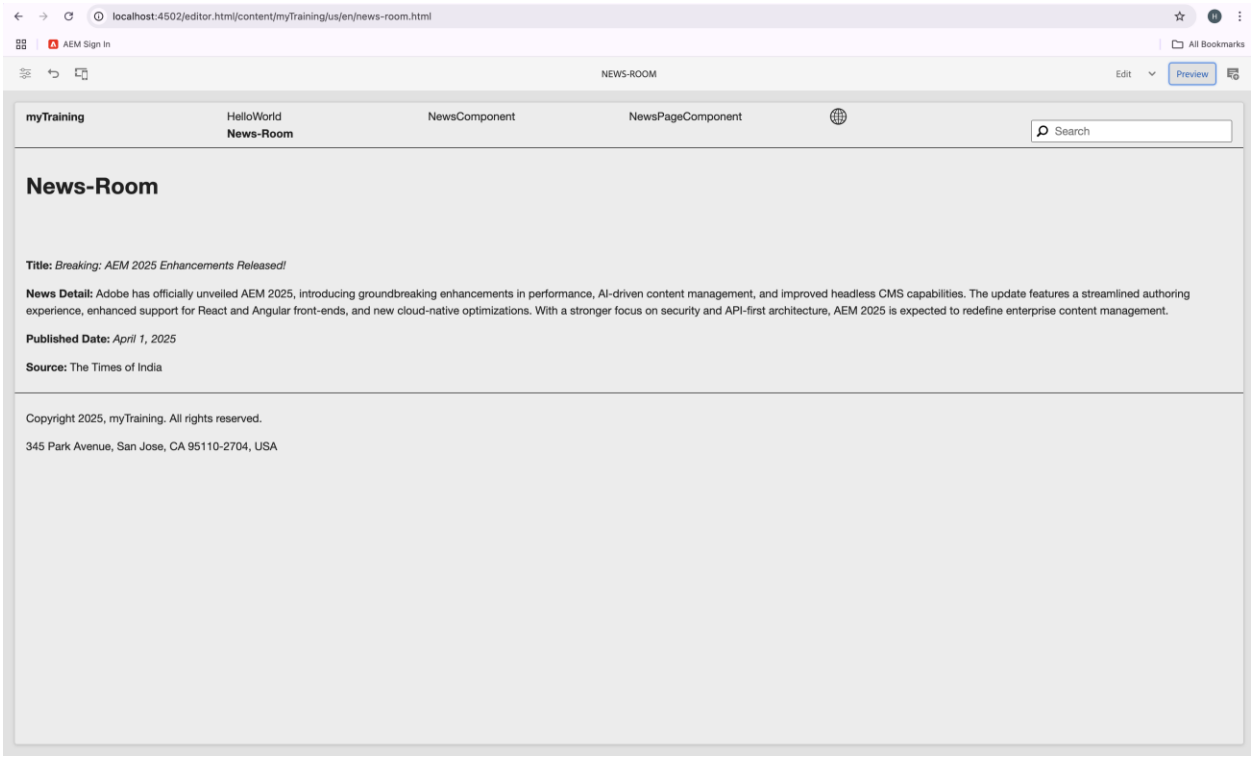
24-05-2025 TASK

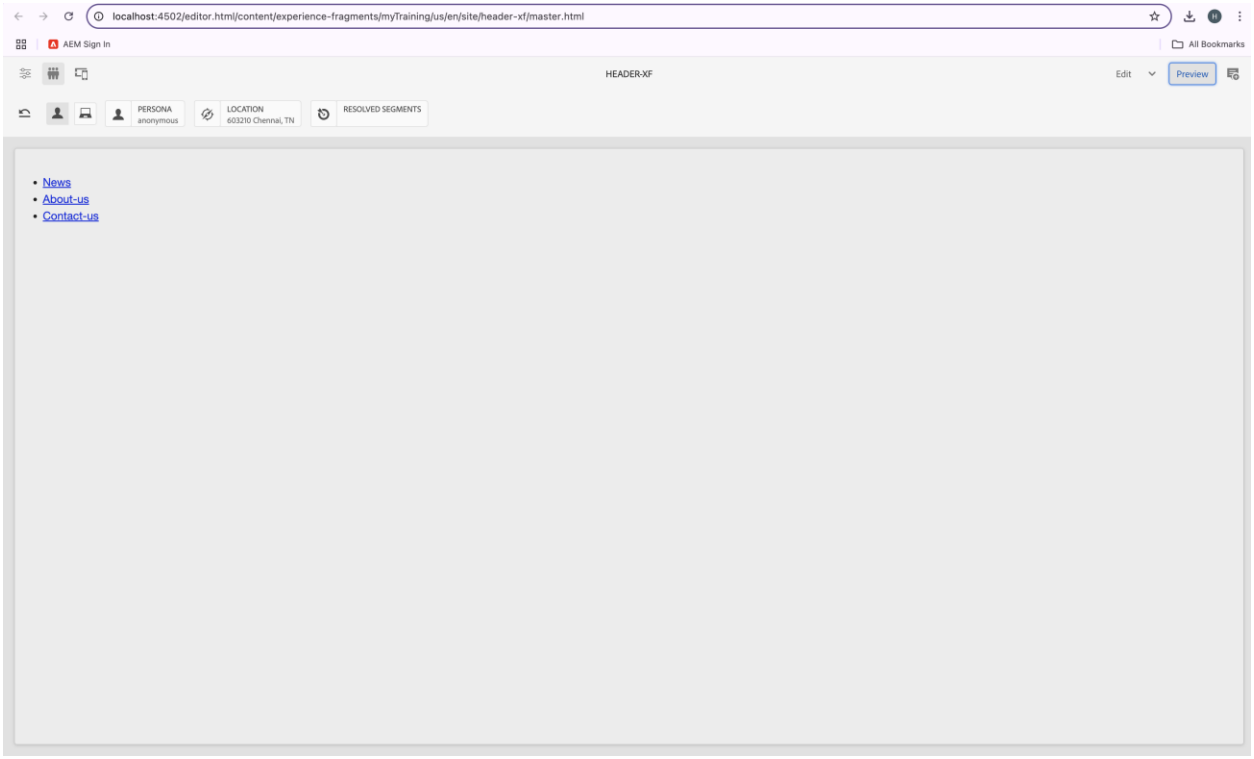
By Harini. A

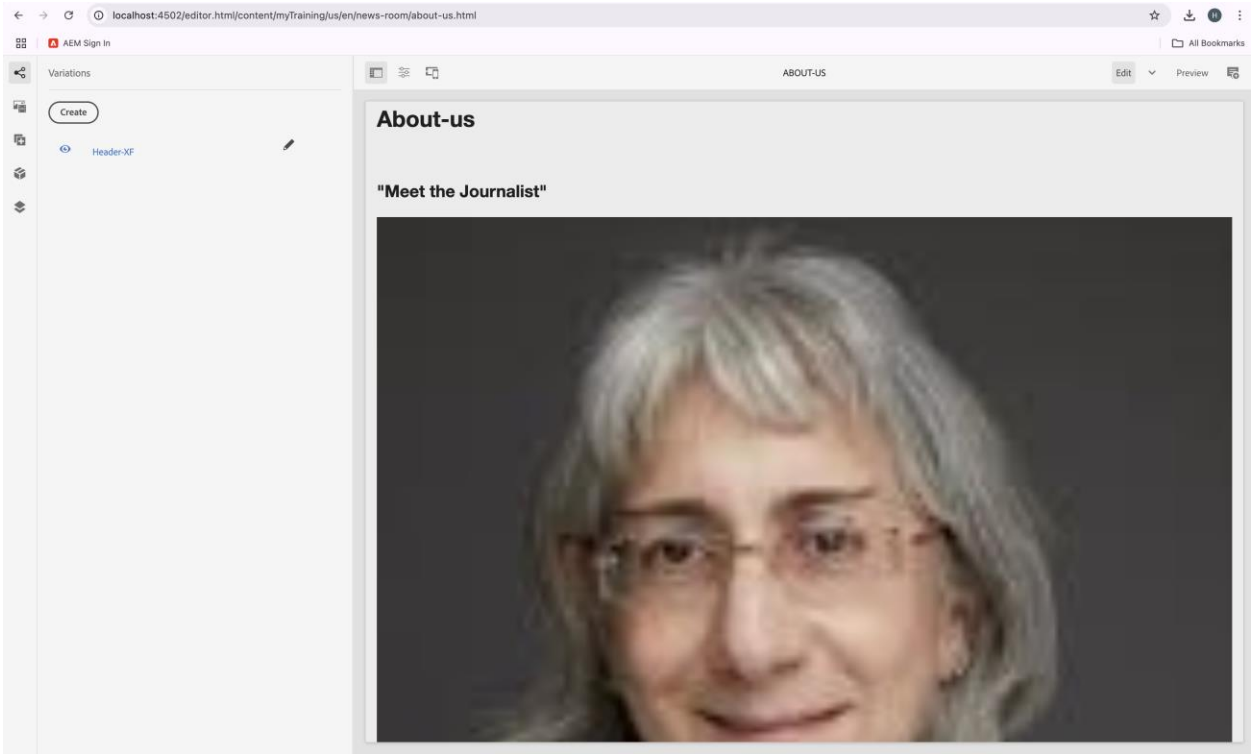
1.Create 5 news article pages under /content/us/en/news All should be unique Use News component which we had created previously to provide the title, news detail and published date.

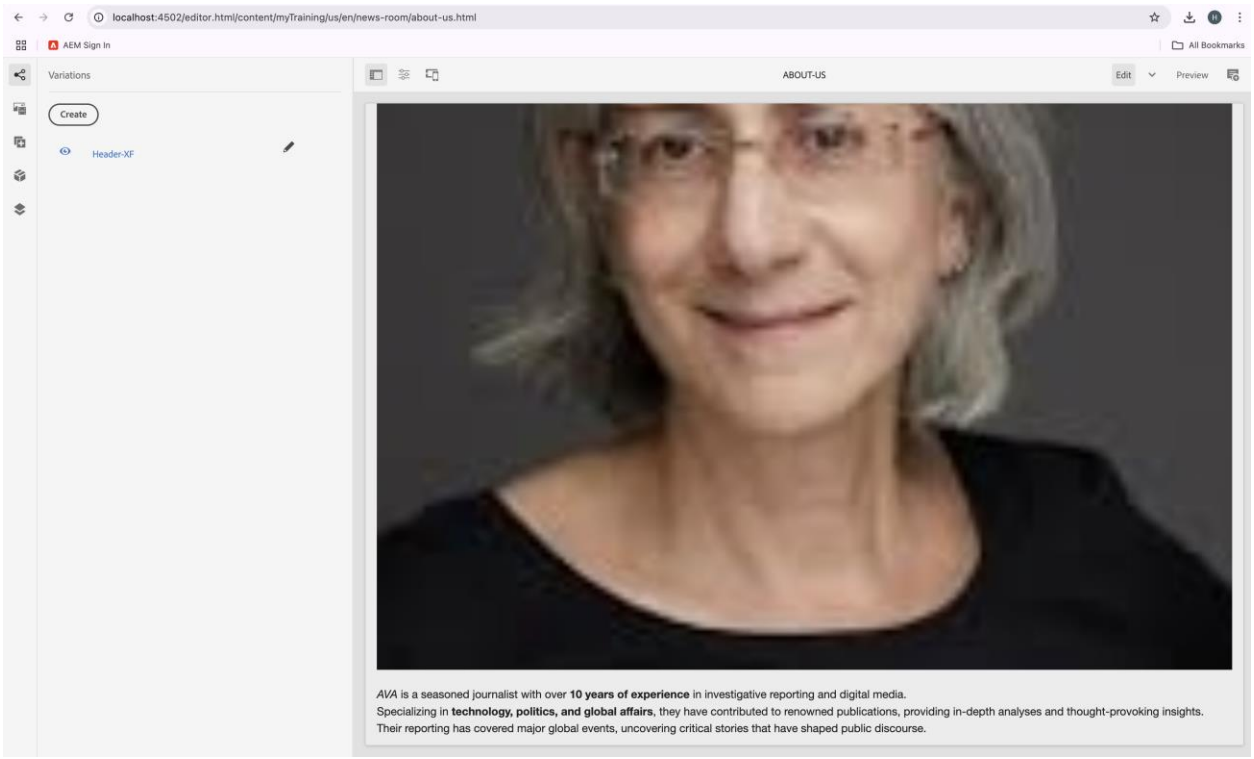


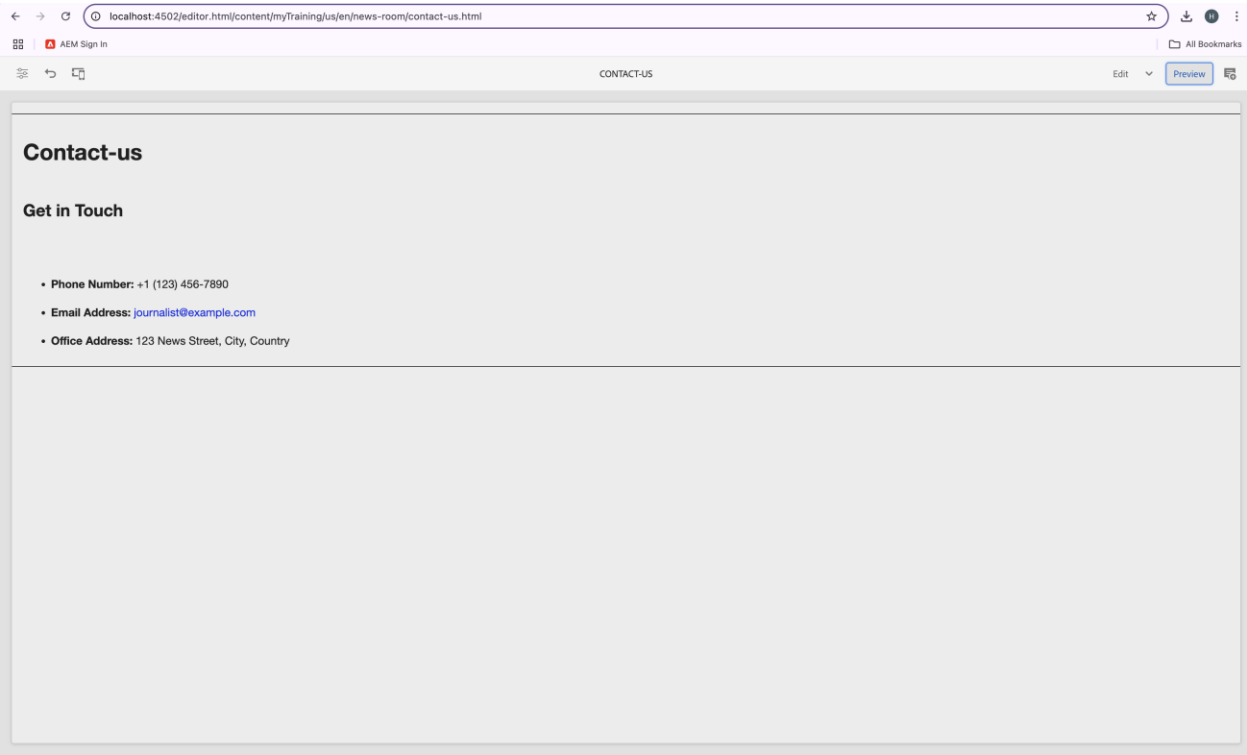


Create Header Experience fragment for header and use these page as menu (name should be news) and apart from hat create contact us page and about me page (you can use teaser or image, text, title components to provide some details about journalist on about me page and on contact us page it should be some contact related details it can have their mobile number office address or email address).




localhost:4502/editor.html/content/myTraining/us/en/news-room/about-us.html
AEM Sign In
All Bookmarks
Variations
Create
Header-XF
ABOUT-US
Edit
Preview
About-us
"Meet the Journalist"


localhost:4502/editor.html/content/myTraining/us/en/news-room/about-us.html
AEM Sign In
All Bookmarks
Variations
Create
Header-XF
ABOUT-US
Edit
Preview
AVA is a seasoned journalist with over **10 years of experience** in investigative reporting and digital media.
Specializing in **technology, politics, and global affairs**, they have contributed to renowned publications, providing in-depth analyses and thought-provoking insights.
Their reporting has covered major global events, uncovering critical stories that have shaped public discourse.



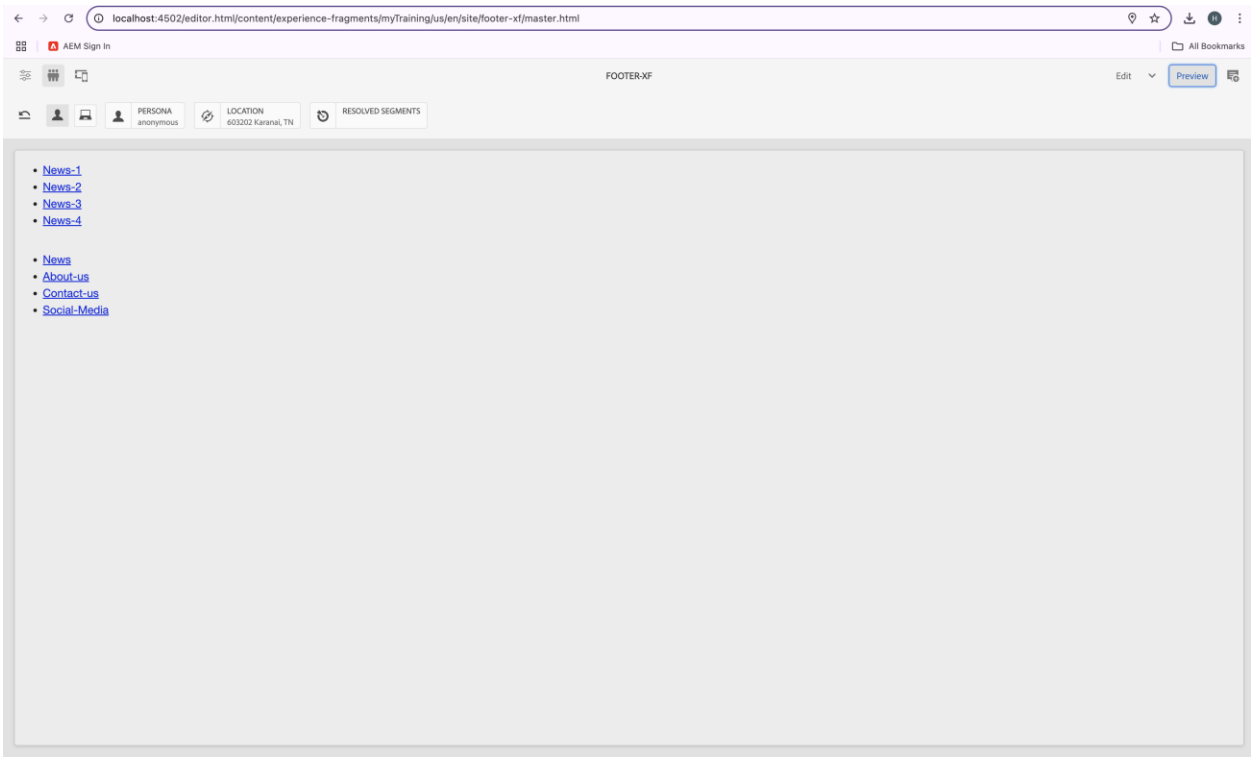
Create footer XF and it could have 4 sections:

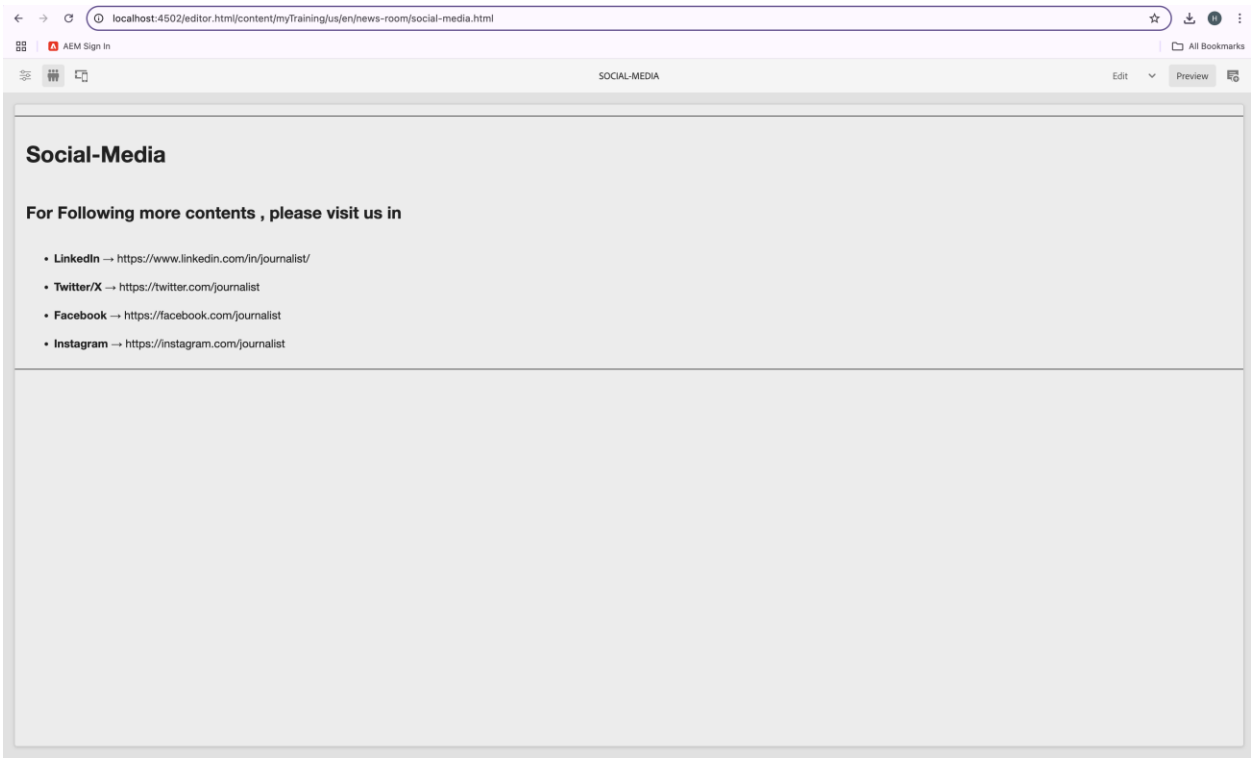
1. News menu section-can have 4 news article you can use list component for this

2. About me section-can have some content and u can use text component

3.contat us section-can have some content and u can use text component

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


localhost:4502/editor.html/content/experience-fragments/myTraining/us/en/site/footer-xf/master.html
AEM Sign In
All Bookmarks
FOOTER-XF
Edit
Preview
PERSONA
anonymous
LOCATION
603202 Karanai, TN
RESOLVED SEGMENTS
News-1
News-2
News-3
News-4
News
About-us
Contact-us
Social-Media


localhost:4502/editor.html/content/myTraining/us/en/news-room/social-media.html
AEM Sign In
All Bookmarks
SOCIAL-MEDIA
Edit
Preview
Social-Media
For Following more contents , please visit us in
LinkedIn → https://www.linkedin.com/in/journalist/
Twitter/X → https://twitter.com/journalist
Facebook → https://facebook.com/journalist
Instagram → https://instagram.com/journalist

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. Navigate to your **AEM Core Module** in the backend project.
2. Inside core/src/main/java/com/example/core/services/, create an interface:

□ HelloWorldService.java

```
package com.example.core.services;
public interface HelloWorldService {
    String getMessage();
}
```

□ HelloWorldServiceImpl.java

```
package com.example.core.services.impl;
import com.example.core.services.HelloWorldService;
import org.osgi.service.component.annotations.Component;
import org.osgi.service.component.annotations.ServiceScope;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

@Component(service = HelloWorldService.class, immediate = true, scope =
ServiceScope.SINGLETON)
public class HelloWorldServiceImpl implements HelloWorldService {
 private static final Logger LOG =
LoggerFactory.getLogger(HelloWorldServiceImpl.class);

    @Override
    public String getMessage() {
        String message = "Hello World from AEM Service!";
        LOG.info("HelloWorldService Message: {}", message);
        return message;
    }
```

```
}
```

□ NewsModel.java

```
package com.example.core.models;

import com.example.core.services.HelloWorldService;
import org.apache.sling.api.resource.Resource;
import org.apache.sling.models.annotations.DefaultInjectionStrategy;
import org.apache.sling.models.annotations.Model;
import org.apache.sling.models.annotations.injectorspecific.Self;
import org.apache.sling.models.annotations.injectorspecific.OSGiService;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import javax.inject.Inject;

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

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

    @OSGiService
    private HelloWorldService helloWorldService;

    @Self
    private Resource resource;

    public String getHelloMessage() {
        String message = helloWorldService.getMessage();
        LOG.info("NewsModel - Received message: {}", message);
        return message;
    }
}
```

□ news.html

```
<div>
    <h2>News Component</h2>
    <p>${newsModel.helloMessage}</p> </div>
```

7.Create custom configurations where I can provide the 3rd party api for exmaple(https://jsonplaceholder.typicode.com/posts) and I can see the data as json. And this data should be print in logs.

 ThirdPartyApiConfig.java

```
package com.example.core.configurations;

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

@ObjectClassDefinition(name = "Third Party API Configuration")
public @interface ThirdPartyApiConfig {

    @AttributeDefinition(name = "API URL", description = "Enter
the API Endpoint")
    String apiUrl() default
"https://jsonplaceholder.typicode.com/posts";
}
```

 ThirdPartyApiService.java

```
package com.example.core.services;

public interface ThirdPartyApiService {
    String fetchData();
}
```

 ThirdPartyApiServiceImpl.java

```
package com.example.core.services.impl;

import com.example.core.configurations.ThirdPartyApiConfig;
import com.example.core.services.ThirdPartyApiService;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;
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.Reference;
import org.osgi.service.metatype.annotations.Designate;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

@Component(service = ThirdPartyApiService.class, immediate =
true)
@Designate(ocd = ThirdPartyApiConfig.class)
public class ThirdPartyApiServiceImpl implements
ThirdPartyApiService {
```

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

    private String apiUrl;

    @Activate
    @Modified
    protected void activate(ThirdPartyApiConfig config) {
        this.apiUrl = config.apiUrl();
        LOG.info("API URL set to: {}", this.apiUrl);
    }

    @Override
    public String fetchData() {
        try (CloseableHttpClient client = HttpClients.createDefault())
{
            HttpGet request = new HttpGet(apiUrl);
            try (CloseableHttpResponse response =
client.execute(request)) {
                String jsonResponse =
EntityUtils.toString(response.getEntity());
                LOG.info("Fetched Data from API: {}", jsonResponse);
                return jsonResponse;
            }
        } catch (Exception e) {
            LOG.error("Error fetching API data: ", e);
        }
        return "{}";
    }
```

```
}
```

 NewsModel.java

```
@OSGiService
private ThirdPartyApiService thirdPartyApiService;

public String getApiData() {
    return thirdPartyApiService.fetchData();
}
```

 news.html

```
<div>
    <h3>Third Party API Data:</h3>
    <pre>${newsModel.apiData}</pre>
</div>
```