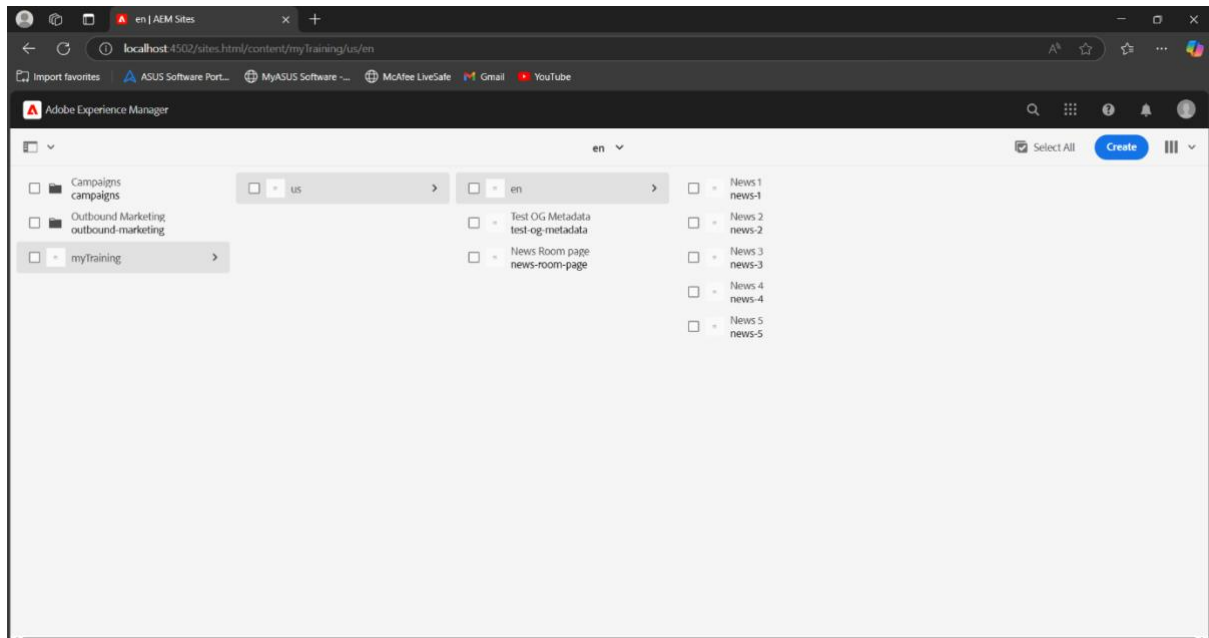
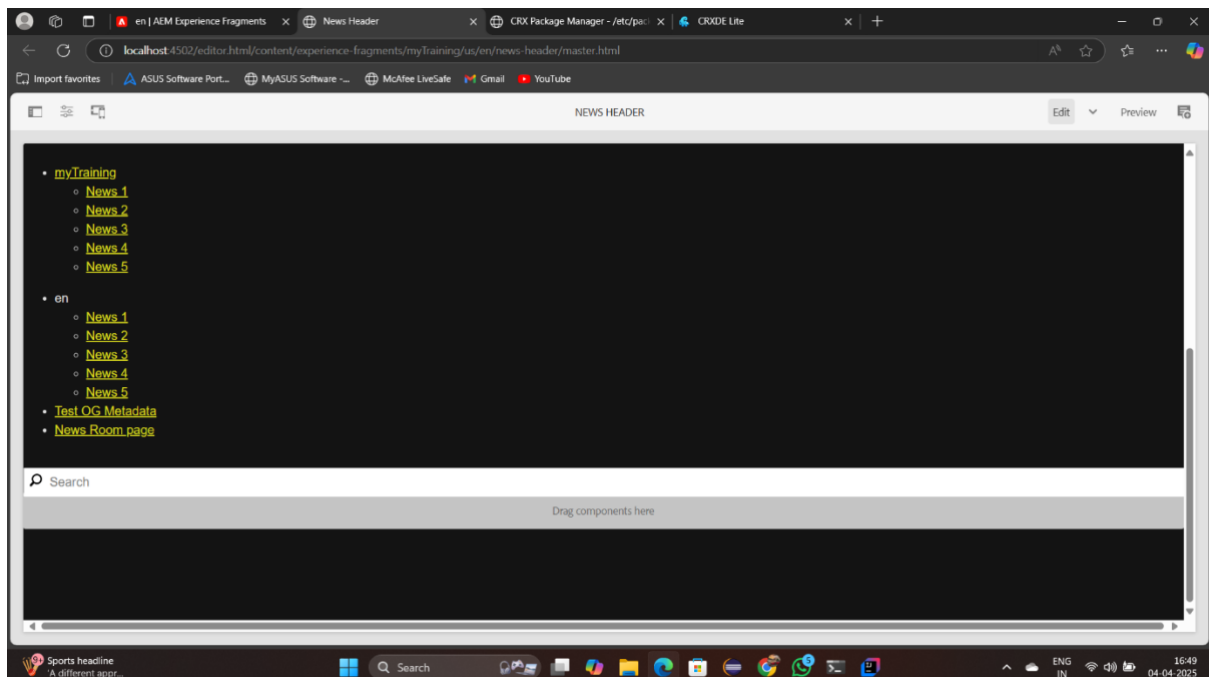


Create 5 news article pages under /content/us/en/news

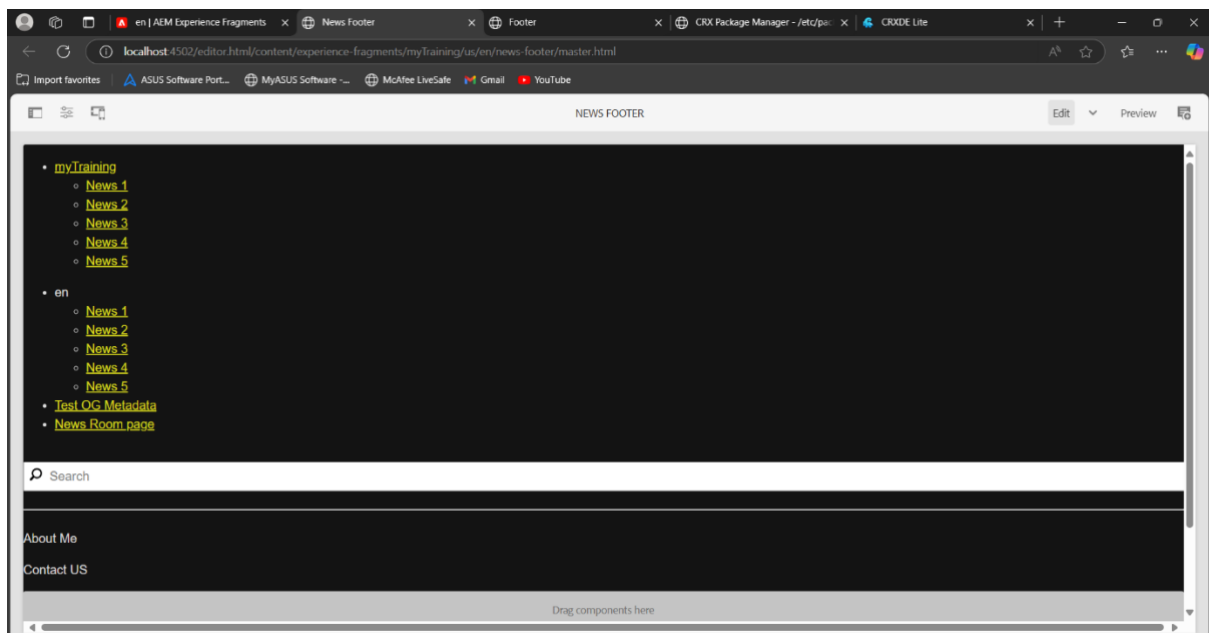
All should be unique ,Use News component which we had created previously to provide the title, news detail and published date



Create Header Experience fragment for header



Create footer XF :



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.Create an OSGi Service Class

package com.myTraining.services;

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

```
@Component(service = HelloWorldService.class)
```

```
public class HelloWorldService {
```

```
    public String getMessage() {
```

```
        return "Hello World from OSGi Service!";
```

```
    }
```

```
}
```

2.Call Service in News Component Sling Model

```
package com.myTraining.models;
```

```
import com.myTraining.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.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;
```

```
    public String getHelloMessage() {
```

```
        String message = helloWorldService.getMessage();
```

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

```
        return message;
```

```
    }
```

```
}
```

Create a Custom Configuration for 3rd-Party API

1. Create OSGi Configuration

```
package com.myTraining.config;
```

```
import org.apache.sling.caconfig.annotation.Configuration;
```

```
import org.apache.sling.caconfig.annotation.Property;
```

```
@Configuration(label = "News API Configuration", description = "Configuration  
for fetching news API data")
```

```
public @interface ApiConfig {
```

```
    @Property(label = "API Endpoint", description = "3rd party API URL")
```

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

```
}
```

2. Create a Service to Fetch Data

```
package com.myTraining.services;
```

```
import com.myTraining.config.ApiConfig;
```

```
import org.apache.sling.caconfig.annotation.Configuration;
```

```
import org.apache.sling.caconfig.management.ConfigurationResolver;
```

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

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

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

```
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.Reference;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;


import java.io.IOException;


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

public class NewsApiService {

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


    @Reference

    private ConfigurationResolver configResolver;


    public String fetchNewsData() {

        String apiUrl = configResolver.get(ApiConfig.class).apiEndpoint();
        try (CloseableHttpClient client = HttpClients.createDefault()) {
            HttpGet request = new HttpGet(apiUrl);
            CloseableHttpResponse response = client.execute(request);
            return EntityUtils.toString(response.getEntity());
        } catch (IOException e) {
            LOG.error("Error fetching news data", e);
            return "Error fetching data";
        }
    }
}
```

3. Call This in News Component Model

```
public String getApiData() {  
    String data = newsApiService.fetchNewsData();  
    LOG.info("API Data: {}", data);  
    return data;  
}
```