

# AJAX AND XML

## AIM :

To create an XML file containing news articles, each with a title, author, publication date, and summary. Use AJAX to fetch and display the articles, and allow users to filter the news articles by date or author.

## PROCEDURE:

### 1.Initialize the Web Page

- Load the index.html file in the browser.
- Include styles.css for styling.
- Link script.js for functionality.

### 2.Load News Articles

- When the page loads, loadNews() fetches the news.xml file.
- The XML file contains structured news data.

### 3.Display News Articles

- Parse the XML and extract title, author, date, and summary.
- Display each article inside newsContainer.

### 4.Filter News Articles

- User enters author name or selects a date.
- filterNews() checks for matching articles and hides non-matching ones.

### 5.Reset Filters

- resetFilters() clears the input fields.
- Reloads all articles by calling loadNews().

### 6.Toggle Dark Mode

- Clicking darkModeToggle adds/removes the .dark-mode class.

- Changes background and text color.

## CODE :

### Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>News Feed</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <button id="darkModeToggle">🌙 Dark Mode</button>
  <h1>📰 Latest News</h1>
  <div class="filters">
    <label>Filter by Author:</label>
    <input type="text" id="authorFilter" placeholder="Enter author name">
    <label>Filter by Date:</label>
    <input type="date" id="dateFilter">
    <button onclick="filterNews()">Filter</button>
    <button onclick="resetFilters()">Reset</button>
    <button onclick="loadNews()">🔄 Refresh News</button>
  </div>
  <div id="newsContainer"></div>
  <script src="script.js"></script>
</body>
```

</html>

**news.xml**

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

<news>

<article>

<title>New AI Breakthrough</title>

<author>John Doe</author>

<date>2025-03-21</date>

<summary>A new AI model has been developed that can learn 10x faster than before.</summary>

</article>

<article>

<title>Economic Boom in 2025</title>

<author>Jane Smith</author>

<date>2025-03-20</date>

<summary>Stock markets are soaring as global economies recover.</summary>

</article>

<article>

<title>Football World Cup Finals</title>

<author>Mike Johnson</author>

<date>2025-03-19</date>

<summary>The championship game was one of the best in history.</summary>

</article>

<article>

<title>SpaceX's New Mission</title>

<author>Elon Musk</author>

<date>2025-03-18</date>

<summary>SpaceX plans to send humans to Mars by 2030.</summary>

</article>

</news>

### **script.js**

```
document.addEventListener("DOMContentLoaded", function () {  
    loadNews();  
  
    document.getElementById("darkModeToggle").addEventListener("click",  
toggleDarkMode);  
});  
  
function loadNews() {  
    let xhr = new XMLHttpRequest();  
    xhr.open("GET", "news.xml", true);  
    xhr.onreadystatechange = function () {  
        if (xhr.readyState == 4 && xhr.status == 200) {  
            displayNews(xhr.responseXML);  
        }  
    };  
    xhr.send();  
}  
  
function displayNews(xml) {  
    let articles = xml.getElementsByTagName("article");  
    let container = document.getElementById("newsContainer");  
    container.innerHTML = "";  
    for (let i = 0; i < articles.length; i++) {  
        let title = articles[i].getElementsByTagName("title")[0].textContent;  
        let author = articles[i].getElementsByTagName("author")[0].textContent;  
        let date = articles[i].getElementsByTagName("date")[0].textContent;  
        let summary = articles[i].getElementsByTagName("summary")[0].textContent;  
        let articleHTML = `
```

```

    <div class="article">
        <h2>${title}</h2>
        <p><strong>Author:</strong> ${author}</p>
        <p><strong>Date:</strong> ${date}</p>
        <p>${summary}</p>
    </div>
`;
    container.innerHTML += articleHTML;
}
}

function filterNews() {
    let authorFilter = document.getElementById("authorFilter").value.toLowerCase();
    let dateFilter = document.getElementById("dateFilter").value;
    let articles = document.querySelectorAll(".article");
    articles.forEach(article => {
        let author = article.querySelector("p:nth-child(2)").textContent.toLowerCase();
        let date = article.querySelector("p:nth-child(3)").textContent;

        if ((authorFilter && !author.includes(authorFilter)) || (dateFilter && date !==
dateFilter)) {
            article.style.display = "none";
        } else {
            article.style.display = "block";
        }
    });
}

function resetFilters() {
    document.getElementById("authorFilter").value = "";

```

```
document.getElementById("dateFilter").value = "";  
loadNews();  
}  
function toggleDarkMode() {  
    document.body.classList.toggle("dark-mode");  
}
```

### style.css

```
body {  
    font-family: Arial, sans-serif;  
    text-align: center;  
    background-color: #f4f4f4;  
    color: #333;  
}  
h1 {  
    margin-top: 20px;  
}  
.filters {  
    margin: 20px;  
}  
.filters input {  
    padding: 8px;  
    margin: 5px;  
}  
.filters button {  
    padding: 8px 12px;  
    background-color: blue;  
    color: white;  
    border: none;
```

```
    cursor: pointer;
}
.filters button:hover {
    background-color: darkblue;
}
#newsContainer {
    max-width: 800px;
    margin: 0 auto;
}
.article {
    background: white;
    padding: 15px;
    margin: 10px 0;
    border-radius: 5px;
    box-shadow: 2px 2px 10px rgba(0, 0, 0, 0.1);
}
.dark-mode {
    background-color: #333;
    color: white;
}
.dark-mode .article {
    background-color: #444;
    color: white;
}
button#darkModeToggle {
    position: absolute;
    top: 10px;
    left: 10px;
```

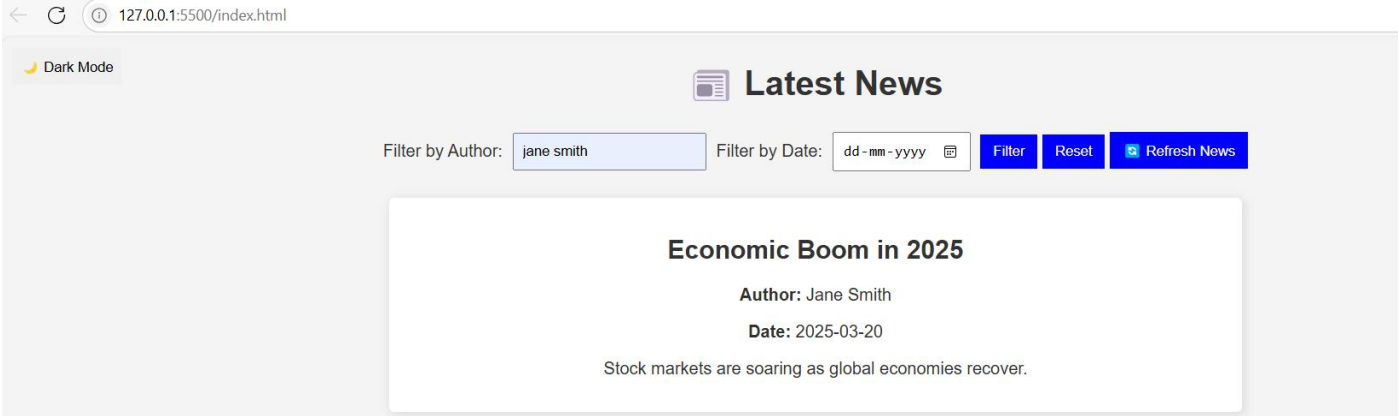
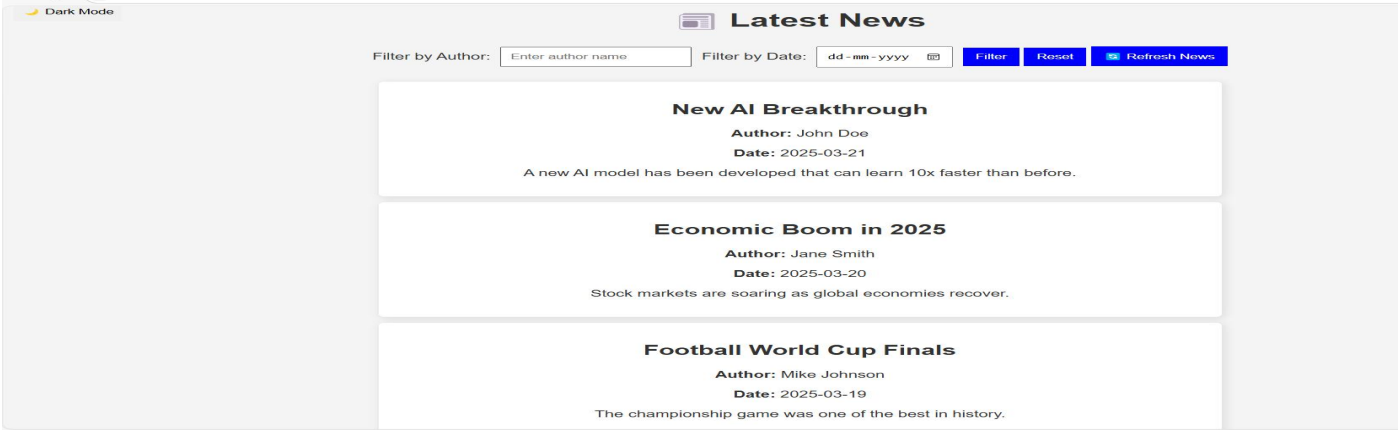
```
padding: 8px;

border: none;

cursor: pointer;

}
```

OUTPUT :



|  |                 |       |
|--|-----------------|-------|
|  | RUBRICES        | MARKS |
|  | OBSERVATION(20) |       |

23CSEC28

VISHALINI S



|                     |  |
|---------------------|--|
| <b>RECORD ( 5 )</b> |  |
| <b>TOTAL ( 25 )</b> |  |
| <b>INITIAL</b>      |  |

### **RESULT :**

Thus, an XML file containing news articles, each with a title, author, publication date, and summary. Use AJAX to fetch and display the articles, and allow users to filter the news articles by date or author are created successfully.