2. Using XPath, perform the following using javascript

- Select All Book Titles:
- Select Authors of Books with Price Less Than 30:
- Select the First Book Title:
- Select All Books with Price Greater Than 25:

```html
<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <title>XML XPath Example</title>
</head>
<body>
 <script type="text/javascript">
  // XML string
  const xmlString = `
  <library>
   <book>
    <title>Introduction to XML</title>
    <author>John Doe</author>
    <price>29.99</price>
   </book>
   <book>
    <title>Web Development Basics</title>
    <author>Jane Smith</author>
    <price>24.95</price>
   </book>
  </library>
  `;

  // Parse the XML string with an alternative method
  const xmlDoc = new DOMParser().parseFromString(xmlString, 'text/xml');

  // Select All Book Titles
  const allBookTitles = xmlDoc.evaluate(
        '//title',                 // XPath expression to select all title elements
        xmlDoc,                    // The XML document to evaluate against
        null,                      // Namespace resolver (not used in this case, set to null)
        XPathResult.ORDERED_NODE_SNAPSHOT_TYPE,  // Result type: ordered snapshot
of nodes
        null                       // Result (not used in this case, set to null)
```

```
        );
    document.write('All Book Titles: ');
    for (let i = 0; i < allBookTitles.snapshotLength; i++) {
      document.write(allBookTitles.snapshotItem(i).textContent + ' ');
    }
    document.write('<br>');

    // Select Authors of Books with Price Less Than 30
    const authorsWithPriceLessThan30 = xmlDoc.evaluate('//book[price < 30]/author', xmlDoc,
null, XPathResult.ORDERED_NODE_SNAPSHOT_TYPE);
    document.write('Authors of Books with Price Less Than 30: ');
    for (let i = 0; i < authorsWithPriceLessThan30.snapshotLength; i++) {
      document.write(authorsWithPriceLessThan30.snapshotItem(i).textContent + ' ');
    }
    document.write('<br>');

    // Select the First Book Title
    const firstBookTitle = xmlDoc.evaluate('//book[1]/title', xmlDoc, null,
XPathResult.STRING_TYPE).stringValue;
    document.write('First Book Title: ' + firstBookTitle + '<br>');

    // Select All Books with Price Greater Than 25
    const booksWithPriceGreaterThan25 = xmlDoc.evaluate('//book[price > 25]', xmlDoc, null,
XPathResult.ORDERED_NODE_SNAPSHOT_TYPE);
    document.write('All Books with Price Greater Than 25: ');
    for (let i = 0; i < booksWithPriceGreaterThan25.snapshotLength; i++) {
      const title =
booksWithPriceGreaterThan25.snapshotItem(i).querySelector('title').textContent;
      const author =
booksWithPriceGreaterThan25.snapshotItem(i).querySelector('author').textContent;
      const price =
booksWithPriceGreaterThan25.snapshotItem(i).querySelector('price').textContent;
      document.write(`${title} ${author} ${price} `);
    }
    document.write('<br>');
  </script>
</body>
</html>
```