Assignment 7 – WhatABook, Part 1

Setup

Each week you will be asked to create a new folder under web-330 following a naming convention of **<week>-number**. If we are on week two, the folder name should be **week-2**. All files associated with the weekly assignment will be added to the appropriate folder. All programs must be linked in the appropriate landing page. Projects will be linked under the "Projects" section of the index.html landing page.

The document title of all HTML files in this course must say "WEB 330 – Enterprise JavaScript II." And, all HTML and CSS files must be valid HTML/CSS, tested through the WC3 validator. The links were provided during WEB 200 and 231. As part of your submission, be sure to include screenshots of the results from the validation tests (HTML and CSS validators).

User interface styling and formatting requirements are located in the Web 330 HTML, CSS, and JavaScript Requirements document.

HTML: **<yourLastName>-whatabook1.html**
CSS:
XML: **books.xml**

Grading Reminders

A. (rubric) All code sources (.html, .css, .js) are measured against
    1. Code functionality: Does it work? Does it meet requirements?
    2. Adherence to standards and conventions. Are you using the appropriate data types, including proper indention, are variables named appropriate (variable x is an example of poor naming conventions), is there an appropriate use of whitespace, is the code organized, and are semicolons being used to terminate code sentences?

3. Efficiency: Use of language features. Are you practicing DRY (Don't-Repeat-Yourself?), are you leveraging built-in language features where appropriate, and are you using classes/functions to reduce code clutter?
4. Documentation: Code is maintainable by others
   i. Code comments are present in all blocks of code, written as full sentences, free of grammatical errors, and function/class parameters and data types have been identified.
   ii. Code attribution is present in all files and authorship is clearly annotated.
5. Error trapping/handling. Are there errors in the program? Is there evidence of coding best practices to reduce user errors?
6. Assignment Specific Compliance. Does the delivered solution follow the instructions, as they are written? Does the output match what was provided in the screenshots (including spaces, styling, etc.)?

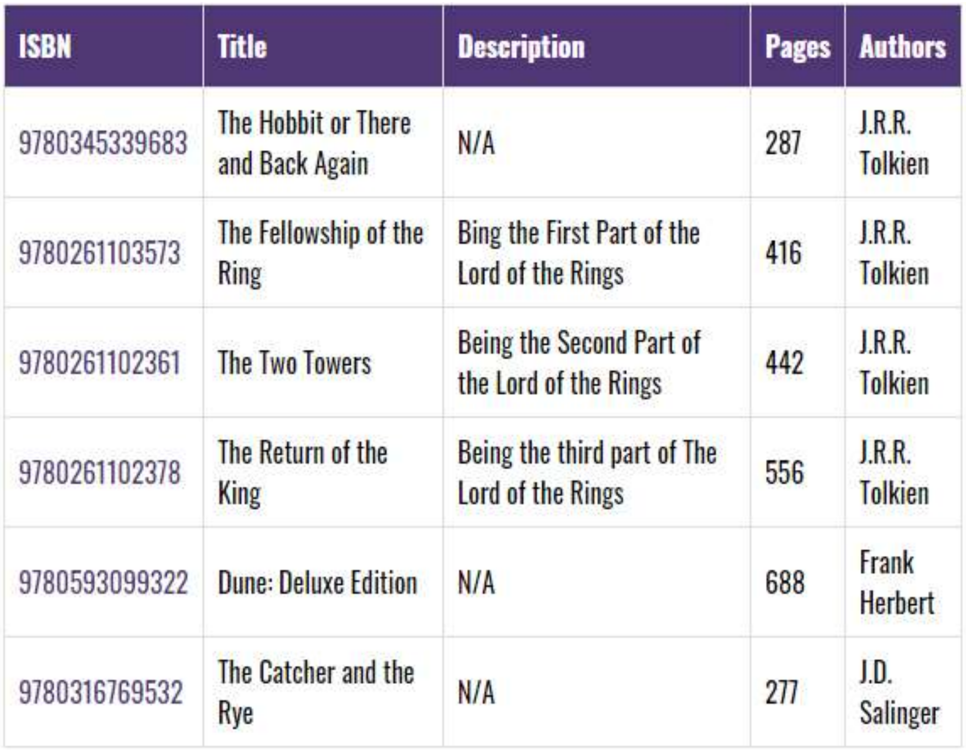
Required Modifications

- Cite any sources in your opening programmer's comment
- Link the appropriate CSS, JavaScript, and Google fonts
- books.xml

Additional Programming Requirements

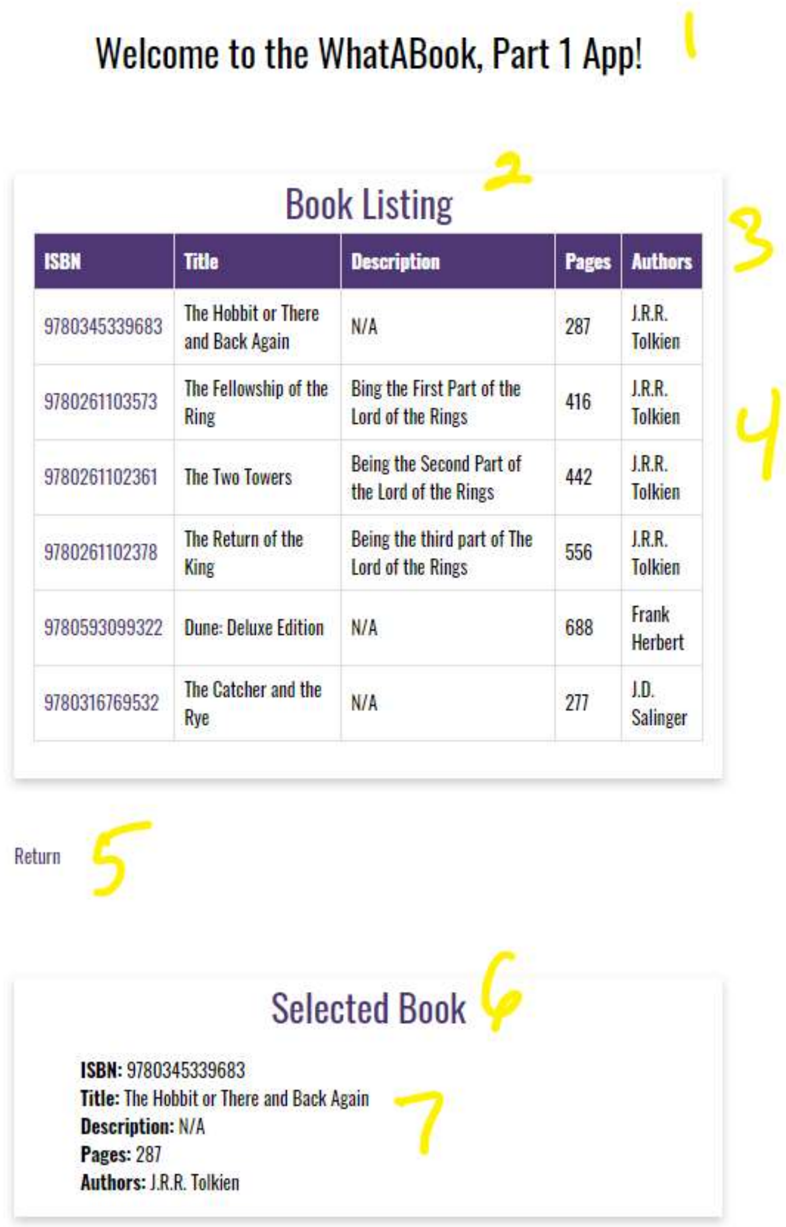
a. Create an XML document with a parent <books> node and child <book> nodes.
b. Each of the child <book> nodes should include the following fields: isbn, title, description, pages, and authors.
c. Populate the XML document with the data from Exhibit A.

Exhibit A. Table data



ISBN	Title	Description	Pages	Authors
9780345339683	The Hobbit or There and Back Again	N/A	287	J.R.R. Tolkien
9780261103573	The Fellowship of the Ring	Bing the First Part of the Lord of the Rings	416	J.R.R. Tolkien
9780261102361	The Two Towers	Being the Second Part of the Lord of the Rings	442	J.R.R. Tolkien
9780261102378	The Return of the King	Being the third part of The Lord of the Rings	556	J.R.R. Tolkien
9780593099322	Dune: Deluxe Edition	N/A	688	Frank Herbert
9780316769532	The Catcher and the Rye	N/A	277	J.D. Salinger

Exhibit B. User Interface (final solution)



Welcome to the WhatABook, Part 1 App!

Book Listing

ISBN	Title	Description	Pages	Authors
9780345339683	The Hobbit or There and Back Again	N/A	287	J.R.R. Tolkien
9780261103573	The Fellowship of the Ring	Bing the First Part of the Lord of the Rings	416	J.R.R. Tolkien
9780261102361	The Two Towers	Being the Second Part of the Lord of the Rings	442	J.R.R. Tolkien
9780261102378	The Return of the King	Being the third part of The Lord of the Rings	556	J.R.R. Tolkien
9780593099322	Dune: Deluxe Edition	N/A	688	Frank Herbert
9780316769532	The Catcher and the Rye	N/A	277	J.D. Salinger

Return

Selected Book

ISBN: 9780345339683
Title: The Hobbit or There and Back Again
Description: N/A
Pages: 287
Authors: J.R.R. Tolkien

1. h1 with a CSS class of app-header and a text value of "Welcome to the WhatABook, Part 1 App!"
2. card-title with a text value of "Book Listing"
3. card-content with an id of bookList (table header)
4. see item 3
5. anchor tag with a link back to the index.html landing page
6. card-title with a text value of "Selected Book"

7. card-content with an id of selectedBook

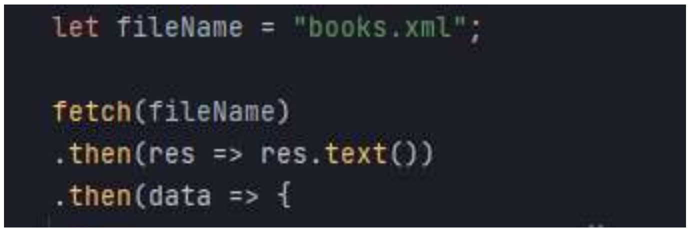
<yourLastName>-whatabook1.html

a. Register an event listener for DOMContentLoaded

Additional JavaScript Requirements

1) Create a variable named fileName and assign it the string value of the books.xml document.
2) Using JavaScripts fetch() API, pass-in the fileName variable.
3) Add a then clause using arrow functions the res data object and call the res.text() function.
4) Add another then clause using arrow functions with the data object (hint: see Exhibit C, but understand you will still need to write the code for the domParser, and xmlBooks variables).

Exhibit C. Fetch API



```
let fileName = "books.xml";

fetch(fileName)
.then(res => res.text())
.then(data => {
```

Additional JavaScript Requirements (body of the second then clause)

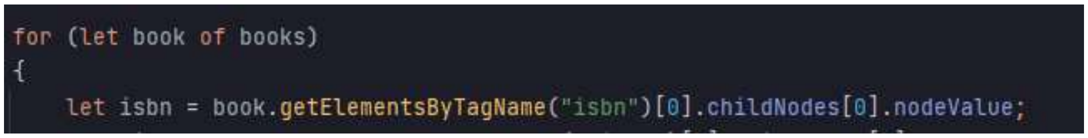
i. Create a new variable named domParser and instantiate a new DOMParser object and assign it to the variable.
ii. Create a new variable named xmlBooks and call the parseFromString() on the domParser variable and supply the function with the data object and the string “text/xml”
iii. Call the laodBooks() function supplying it the xmlBooks variable.
iv. Call the addIsbnClickEvents() function.

b. Create a function named laodBooks() with one parameter with a value of xml.

Additional JavaScript Requirements

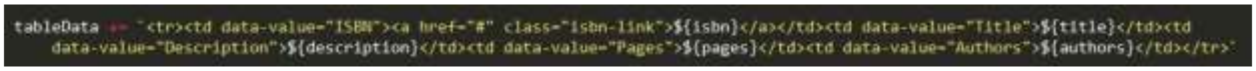
1) Create a variable named books and assign it the results from the xml.getElementsByTagName(“book”) function call.
2) Create a variable named tableData and build the header for an HTML table using the CSS id "bookTable" and CSS class "table"
3) Using a for…of statement, iterate over the books variable and append each XML object to the bookTable variable. See Exhibit D. for a code snippet to help get you started. Pay close attention to Exhibit E. attributes for data-value have been added to each of the td sections. This is need to show the labels in the selected output section (see Exhibit B, item 7). We access this attribution in Exhibit G. with the code snippet **field.dataset.value**

Exhibit D. Starter code snippet, part 1



```
for (let book of books)
{
    let isbn = book.getElementsByTagName("isbn")[0].childNodes[0].nodeValue;
```

Exhibit E. Starter code snippet, part 2



```
tableData += `<tr><td data-value="ISBN"><a href="#" class="isbn-link">${isbn}</a></td><td data-value="Title">${title}</td><td
    data-value="Description">${description}</td><td data-value="Pages">${pages}</td><td data-value="Authors">${authors}</td></tr>`
```

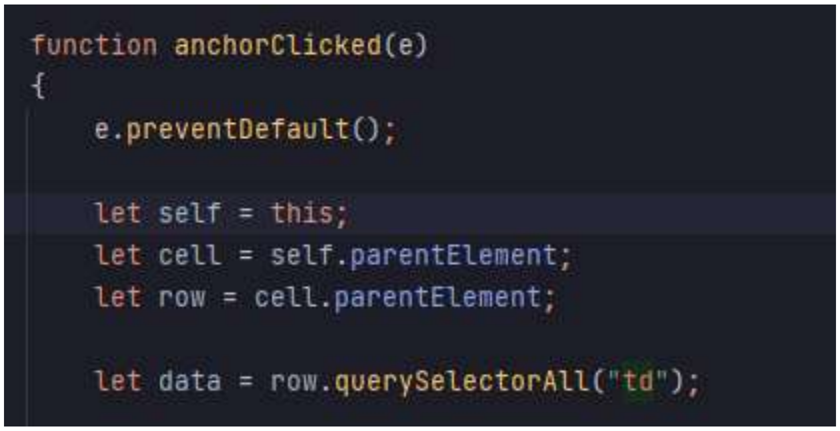
4) Outside of the for…of statement close the HTML table string and bind the variable to the bookList div's innerHTML.

c. Create a function named anchorClicked() with a single parameter of the value e.

Additional JavaScript Requirements

1) Apply the code in Exhibit F. to the function.

Exhibit F.



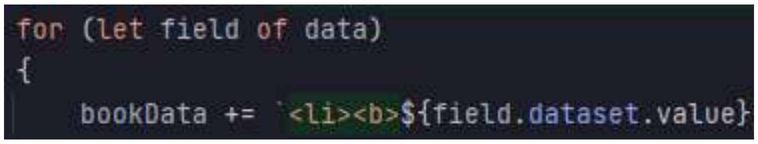
```
function anchorClicked(e)
{
    e.preventDefault();

    let self = this;
    let cell = self.parentElement;
    let row = cell.parentElement;

    let data = row.querySelectorAll("td");
```

2) Next, build an HTML string for an unordered list (set an inline CSS style to list-style-type: none) and use a for…of loop to iterate over the data variable and append the iterated object dataset fields to the unordered list. See Exhibit G (hint: Exhibit G. is a partial view of the solution; you will need to figure out the missing code).

Exhibit G.



```
for (let field of data)
{
    bookData += `<li><b>${field.dataset.value}
```

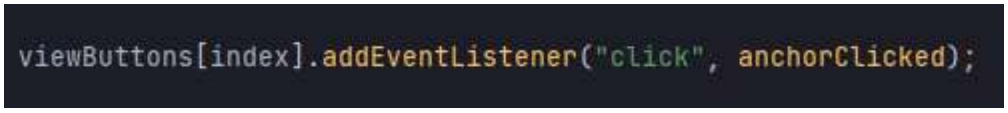
3) Outside for…of statement, bind the bookData variable to the selectedBook.innerHTML property.

d. Create a function named addIsbnClickEvents()

Additional JavaScript Requirements

1) Create a variable named viewButtons and assign it the results from document.querySelectorAll("#bookTable tbody .isbn-link").
2) Create a standard for loop and in the body add event listeners for "click" using the anchorClicked() function. See Exhibit H.

Exhibit H.



```
viewButtons[index].addEventListener("click", anchorClicked);
```

e. Intended functionality: when <yourLastName>-whatabook1.html loads the XML records should appear in an HTML table. If a user selects the ISBN number of an individual book object, the details should appear in the selectedBook section. See Exhibit B.