**WEEK-5**

**AIM:** **Write an XML file which will display the Book information which includes the following: 1) Title of the book 2) Author Name 3) ISBN number 4) Publisher name 5) Edition 6) Price**

**Write a Document Type Definition (DTD) to validate the above XML file. Display the XML file as follows. The contents should be displayed in a table. The header of the table should be in color GREY. And the Author names column should be displayed in one color and should be capitalized and in bold. Use your own colors for remaining columns. Use XML schemas XSL and CSS for the above purpose. Note: Give at least for 4 books. It should be valid syntactically. Hint: You can use some xml editors like XML-spy**

**Procedure:**

1. Declaration:

<?xml version="1.0" encoding="UTF-8"?> declares the document as an XML document, specifying the version (1.0) and character encoding (UTF-8).

2. Processing instruction:

<?xml-stylesheet type="text/xsl" href="book.xsl"?> instructs the processing software (like an XSL processor) to use the external stylesheet file "book.xsl" to transform the XML data. Stylesheets are used to format and present the data in different ways, such as generating HTML, text, or other formats.

3. Document type definition (DTD):

<!DOCTYPE bookstore SYSTEM "book.dtd"> would declare a Document Type Definition (DTD) file named "book.dtd" that defines the structure and element types allowed in this XML document.

4. Root element:

<bookstore> defines the root element of the document, indicating that the content describes a bookstore.

5. Book elements:

Each <book> element represents information about a single book in the bookstore.

<title>...</title> specifies the title of the book.

<author>...</author> specifies the book's author.

<ISBN>...</ISBN> stores the International Standard Book Number (ISBN) for the book.

<publisher>...</publisher> specifies the book's publisher.

<edition>...</edition> indicates the edition number of the book.

<price>...</price> stores the price of the book.

This code represents an XSL stylesheet that transforms the XML data from the previous example (book.xml) into an HTML table. Here's a breakdown of the procedures involved:

**1. HTML structure:**

* The template creates an HTML document structure, including:
  + <html> element defines the root element of the HTML document.
  + <head> element contains information about the document, including:
    - <link rel="stylesheet" href="book.css"/> link to an external CSS file named "book.css" that would style the generated HTML table (assuming the file exists).
  + <body> element contains the visible content of the webpage.

**2. Table creation:**

* Within the <body>, an <h1> element displays the title "My books Collection".
* A table element (<table>) is created with a border (border="1").
* A header row (<tr>) is created with a gray background (bgcolor="grey") containing the table headers (<th>) for "title", "author", "ISBN", "publisher", "edition", and "price".
* The <xsl:for-each> element iterates through each <book> element within the <bookstore> element of the XML document.
  + For each book, a new table row (<tr>) is created.
  + Within the row, individual table cells (<td>) are created for each book element (title, author, ISBN, publisher, edition, and price).
  + The xsl:value-of element retrieves the corresponding data from the XML document using XPath expressions (e.g., select="title" to access the title of the current book).
  + Each cell also has a class assigned (e.g., class="title") which can be used for further styling in the CSS file.

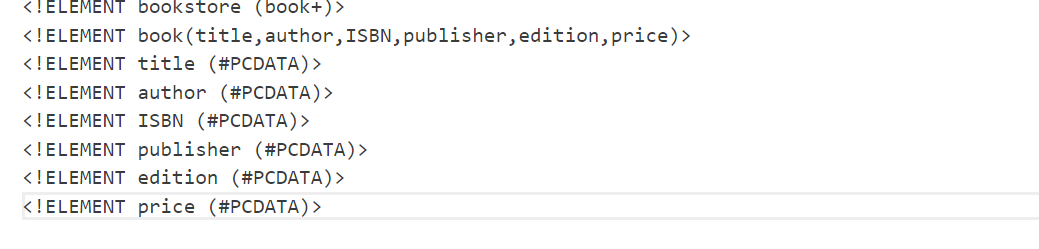
**Source code:**

Bookstore.xml:

****

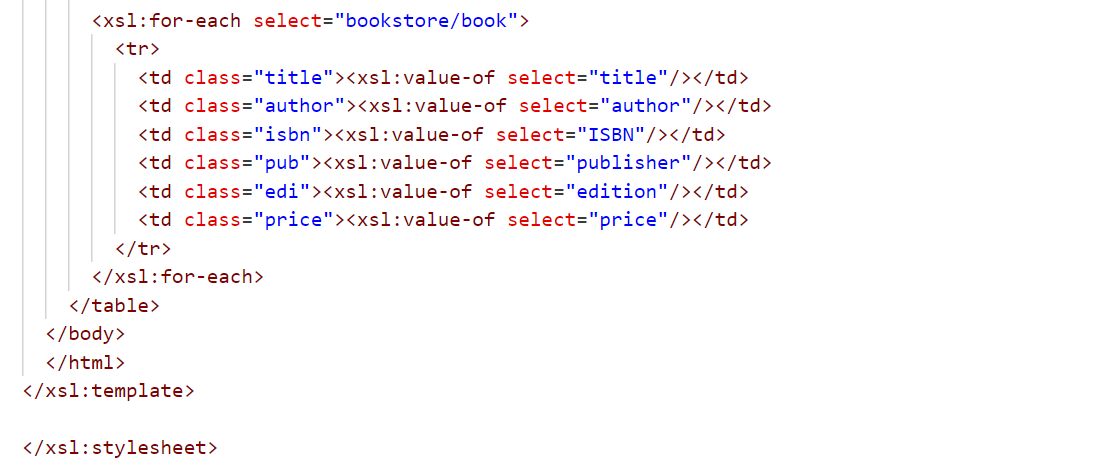
****

Book.dtd:

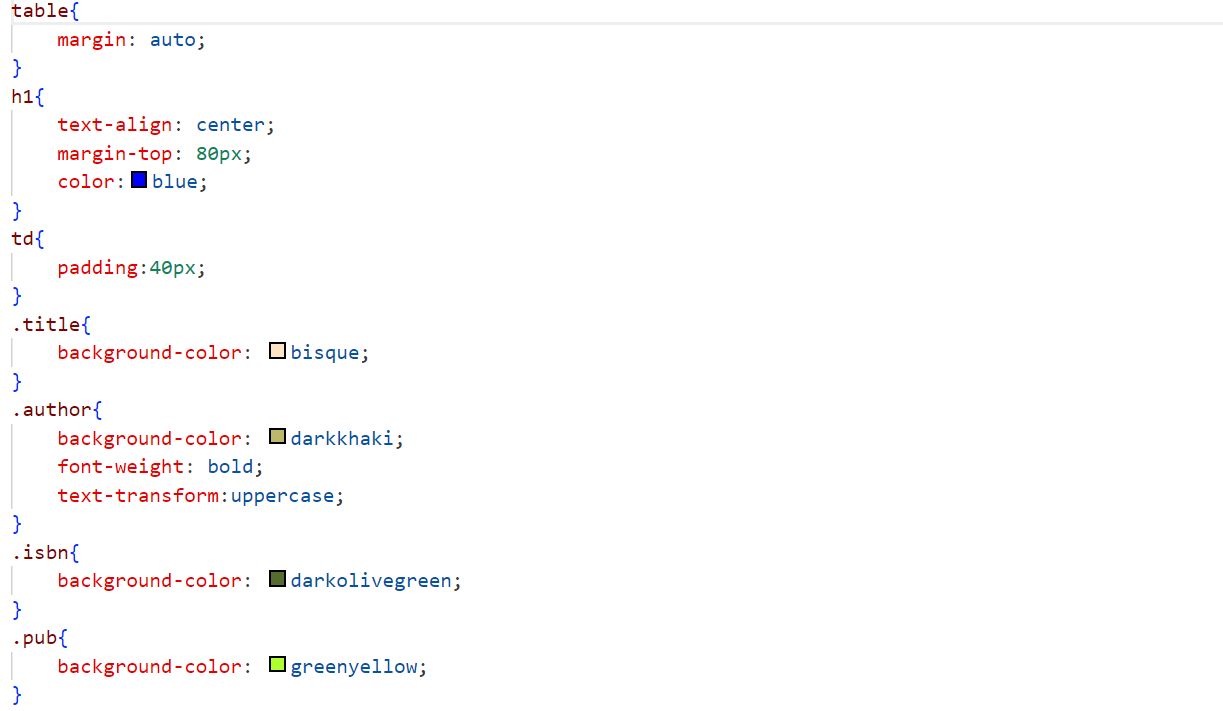


Book.xsl:



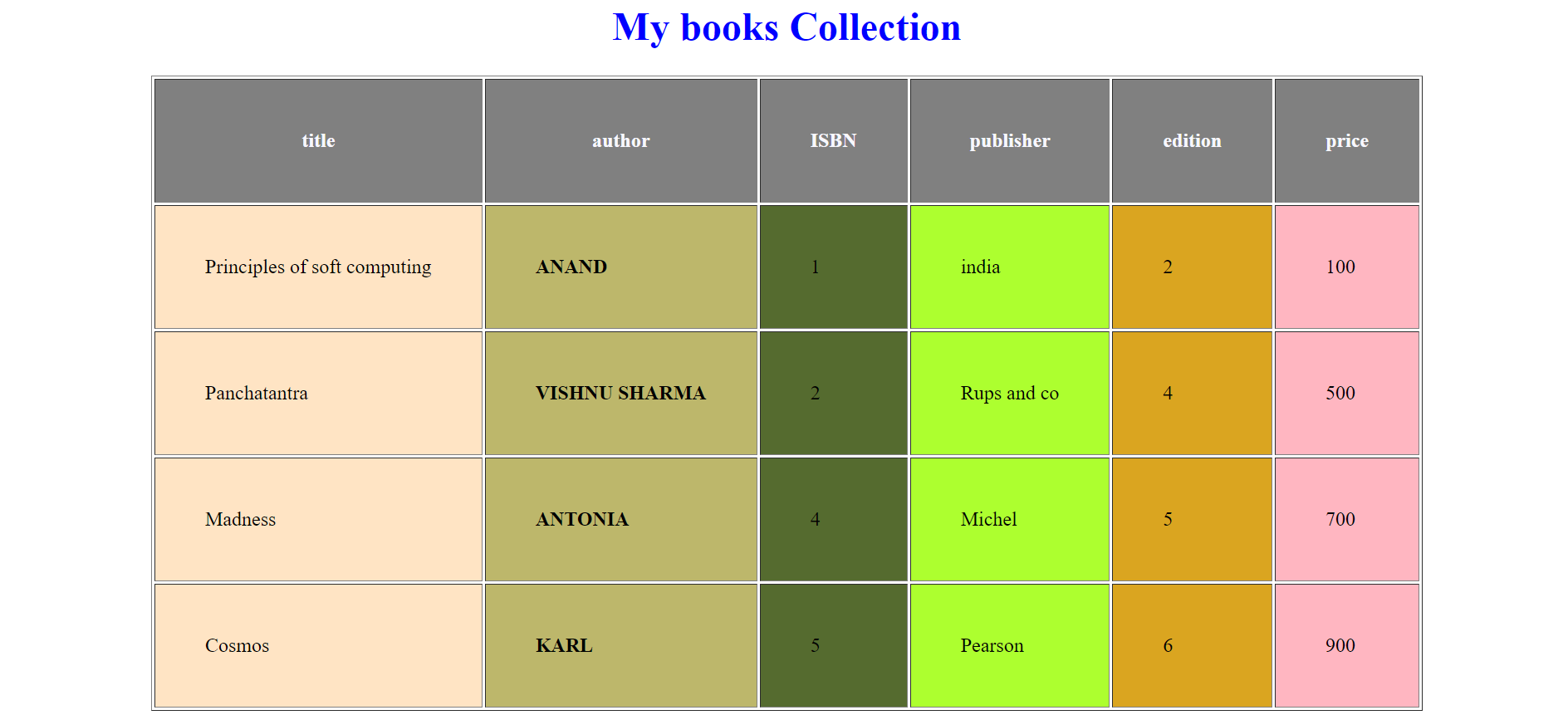


**Book.css:**

****

****

**Output:**

****