Unit 4

Introduction to XML

- XML stands for extensible Markup Language derived from Standard Generalized Markup Language (SGML).
- **XML (Extensible Markup Language)** is a text-based format designed to store and transport data in a way that is both human and machine-readable.
- Unlike HTML, which defines the presentation of data, XML focuses on what the data is.
- It allows users to create custom tags tailored to their specific needs, making it highly flexible for a wide variety of applications.

characteristics of XML

- XML is extensible: XML allows you to create your own self-descriptive tags that suits your application.

- XML carries the data, does not present it: XML allows you to store the data irrespective of how it will be presented.

- XML is a public standard: XML was developed by an organization called the World Wide Web Consortium (W3C) and is available as an open standard.

Benefits of Working with XML

- **Platform Independence:** XML is plain text, meaning it can be created and read on any operating system.
- **Data Sharing:** Its self-descriptive format makes it easy to share and exchange data across different systems and applications.
- **Extensibility:** You can define your own tags and structure, which is ideal for representing complex data structures.
- **Human Readability:** The structure of XML makes it easy for humans to read and understand, facilitating debugging and collaboration.

	HTML	XML
1	It has an extension of .html and .htm	It has an extension of .xml
2	HTML stands for Hyper Text Markup Language.	XML stands for Extensible Markup Language.
3	HTML is static in nature.	XML is dynamic in nature.
4	HTML is not Case sensitive.	XML is Case sensitive.
5	HTML is used to display the data.	XML is used to store data.
6	HTML tags are predefined tags.	XML tags are user-defined tags.
7	It is termed as a presentation language.	It is neither termed as a presentation nor a programming language.
8	In HTML, closing tags are not necessary.	In XML, closing tags are necessary.

Basic Structure of an XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<library>
  <book id="001">
    <title>Learning XML</title>
    <author>John Smith</author>
    <year>2020</year>
  </book>
  <book id="002">
    <title>Advanced XML</title>
    <author>Jane Doe</author>
    <year>2021</year>
  </book>
</library>
```

An XML document is made up of elements, which are the building blocks of the document. Every XML document has a single root element that encloses all other elements.

- **XML Declaration:**
  <?xml version="1.0" encoding="UTF-8"?>
  This declaration is optional but recommended. It specifies the XML version and the character encoding used in the document.

- **Root Element:**
  <library> is the root element that contains all other elements.

- **Child Elements:**
  Each <book> element is a child of <library>. The id attribute uniquely identifies each book.

- **Nested Elements:**
  Inside each <book> element, there are additional elements like <title>, <author>, and <year> that describe the properties of the book.

XML Elements and Attributes

Elements:

Elements are defined by start and end tags. They can contain text, other elements, or both. For example:

<greeting>Hello, world!</greeting>

Attributes:

Attributes provide additional information about elements. They appear within the start tag. For instance:

<user id="123" role="admin">Alice</user>

Here, id and role are attributes that give extra context about the <user> element.

Comments in XML

- Comments in XML are similar to those in HTML and are used to insert notes that are not processed by XML parsers

<!-- This is a comment in XML -->

You can validate the xml files using following website .You paste your xml code for validating

https://www.xmlvalidation.com/

- Books.XML
- <?xml version="1.0" encoding="UTF-8"?>
- <?xml-stylesheet type="text/css" href="Rule.css"?>
- <books>
- <heading>Welcome To GeeksforGeeks </heading>
- <book>
- <title>Title -: Web Programming</title>
- <author>Author -: Chrisbates</author>
- <publisher>Publisher -: Wiley</publisher>
- <edition>Edition -: 3</edition>
- <price> Price -: 300</price>
- </book>
- <book>
- <title>Title -: Internet world-wide-web</title>
- <author>Author -: Ditel</author>
- <publisher>Publisher -: Pearson</publisher>
- <edition>Edition -: 3</edition>
- <price>Price -: 400</price>
- </book>
- <book>
- <title>Title -: Computer Networks</title>
- <author>Author -: Foruouzan</author>
- <publisher>Publisher -: Mc Graw Hill</publisher>
- <edition>Edition -: 5</edition>
- <price>Price -: 700</price>
- </book>

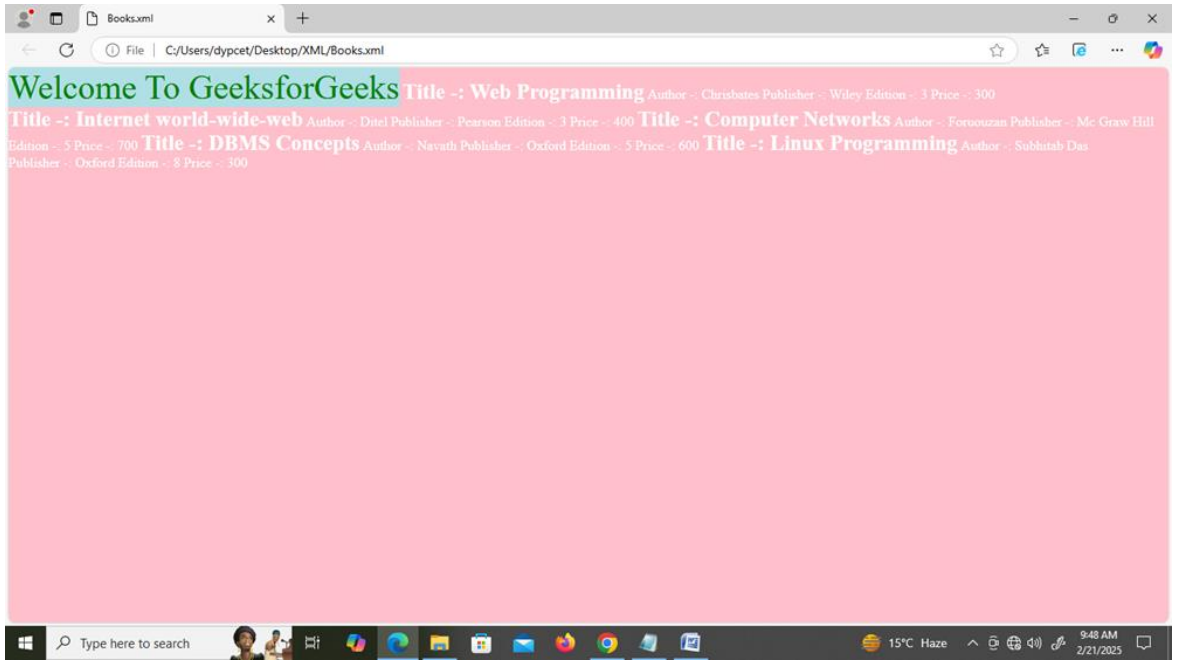
<book>
<title>Title -: DBMS Concepts</title>
<author>Author -: Navath</author>
<publisher>Publisher -: Oxford</publisher>
<edition>Edition -: 5</edition>
<price>Price -: 600</price>
</book>
<book>
<title>Title -: Linux Programming</title>
<author>Author -: Subhitab Das</author>
<publisher>Publisher -: Oxford</publisher>
<edition>Edition -: 8</edition>
<price>Price -: 300</price>
</book>
</books>

Rule.css

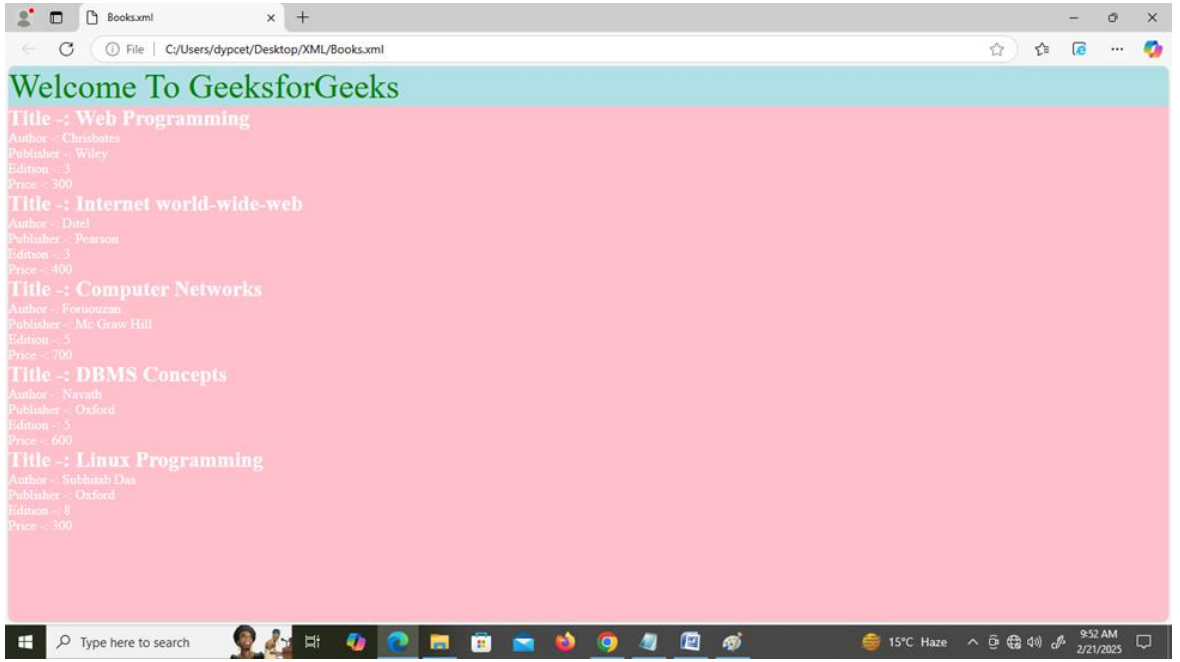
```
books {
        color: white;
        background-color : pink;
        width: 100%;
}
heading {
        color: green;
        font-size : 40px;
        background-color : powderblue;
}
heading, title, author, publisher, edition {
        display : inline;
}
title {
        font-size : 25px;
        font-weight : bold;
}
```

Rule1.css
If we connect blank Rule1.css to Books.xml then it will display only data of XML

Output1 display as inline



Output2 display as block



Value	Description
inline	Displays an element as an inline element (like). Any height and width properties will have no effect. This is default.
block	Displays an element as a block element (like <p>). It starts on a new line, and takes up the whole width
contents	Makes the container disappear, making the child elements children of the element the next level up in the DOM
flex	Displays an element as a block-level flex container
grid	Displays an element as a block-level grid container
inline-block	Displays an element as an inline-level block container. The element itself is formatted as an inline element, but you can apply height and width values
inline-flex	Displays an element as an inline-level flex container
inline-grid	Displays an element as an inline-level grid container
inline-table	The element is displayed as an inline-level table
list-item	Let the element behave like a element
run-in	Displays an element as either block or inline, depending on context
table	Let the element behave like a <table> element
table-caption	Let the element behave like a <caption> element
table-column-group	Let the element behave like a <colgroup> element
table-header-group	Let the element behave like a <thead> element
table-footer-group	Let the element behave like a <tfoot> element
table-row-group	Let the element behave like a <tbody> element
table-cell	Let the element behave like a <td> element
table-column	Let the element behave like a <col> element
table-row	Let the element behave like a <tr> element
none	The element is completely removed
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit