Reg.No:22IT068

Ex No : 8
Date : 10.09.24 Practice with XML DTD

QUESTIONS

1. Create an XML file to store details of students (name, age, course, grade). Ensure that the file has at least 3 entries.
    a. How can you modify the structure to include optional elements like address or phone number?

2. Write a DTD for an XML file that defines a bookstore.
    a. The bookstore should contain elements for <book>, with child elements like <title>, <author>, <year>, and <price>.
    b. Ensure that every book has a title and author, but price is optional.

3. Create an XML document representing a company's employee database. Use DTD to ensure that each employee element contains a name, ID, department, and an optional address.
    a. What happens if you try to add an invalid or missing element in the XML file?

4. Write a DTD that validates an XML document for a recipe book. Each recipe should contain a title, an ingredient list, and instructions. Ingredients can have optional attributes like quantity and unit.
    a. How can you enforce that each recipe contains at least one ingredient and one instruction?

5. Using internal DTD, create an XML document that represents a catalog of movies, with elements for title, director, genre, and release year.
    a. How would you enforce that the release year is a four-digit number using DTD?

6. Design an XML and corresponding DTD to represent a product inventory system. Each product must have a unique identifier, name, category, and price. Optionally, it may have a description and stock quantity.
    a. How can you make the unique identifier a required attribute using DTD?

7. Modify an XML document to use an external DTD instead of an internal one.
    a. What changes are necessary in the XML file to reference the external DTD?

AIM

To practice XML along with DTD from the given questions.

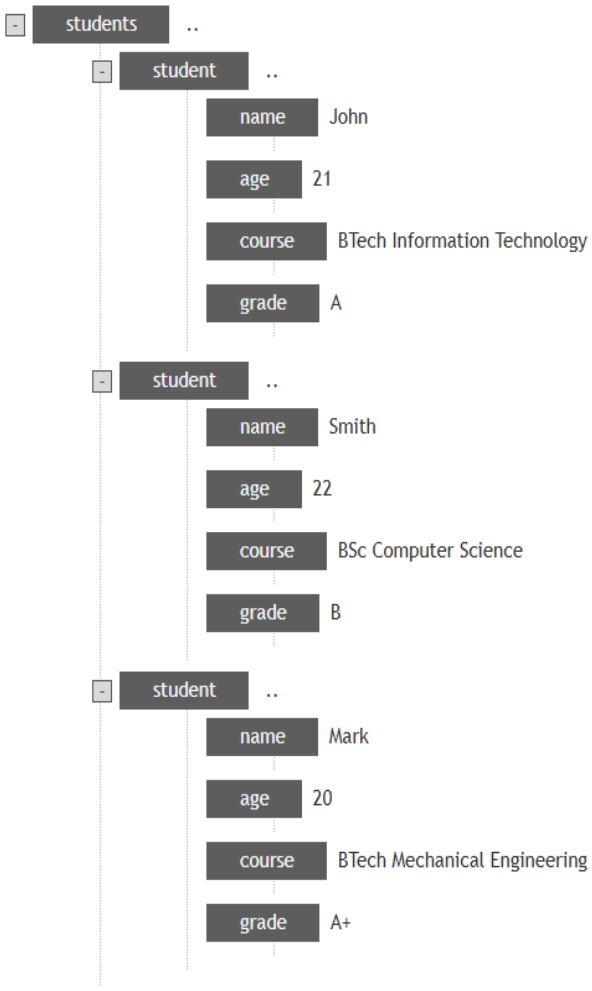
CODE

1.

```
<?xml version="1.0" encoding="UTF-8"?>
<students>
  <student>
    <name>John</name>
    <age>21</age>
    <course>BTech Information Technology</course>
    <grade>A</grade>
  </student>
```

```
    <student>
        <name>Smith</name>
        <age>22</age>
        <course>BSc Computer Science</course>
        <grade>B</grade>
    </student>
    <student>
        <name>Mark</name>
        <age>20</age>
        <course>BTech Mechanical Engineering</course>
        <grade>A+</grade>
    </student>
</students>
```

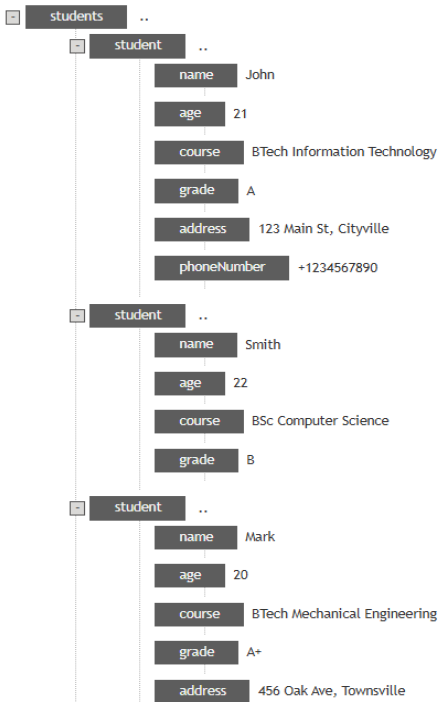
OUTPUT



A. To include optional elements like address or phone number in the XML structure, you can add them as additional child elements within the <student> element. If a student doesn't have an address or phone number, you can simply omit these elements for that particular student.

```
<?xml version="1.0" encoding="UTF-8"?>
<students>
  <student>
    <name>John</name>
    <age>21</age>
    <course>BTech Information Technology</course>
    <grade>A</grade>
    <address>123 Main St, Cityville</address>
    <phoneNumber>+1234567890</phoneNumber>
  </student>
  <student>
    <name>Smith</name>
    <age>22</age>
    <course>BSc Computer Science</course>
    <grade>B</grade>
    <!-- No address or phone number -->
  </student>
  <student>
    <name>Mark</name>
    <age>20</age>
    <course>BTech Mechanical Engineering</course>
    <grade>A+</grade>
    <address>456 Oak Ave, Townsville</address>
    <!-- No phone number -->
  </student>
</students>
```

OUTPUT

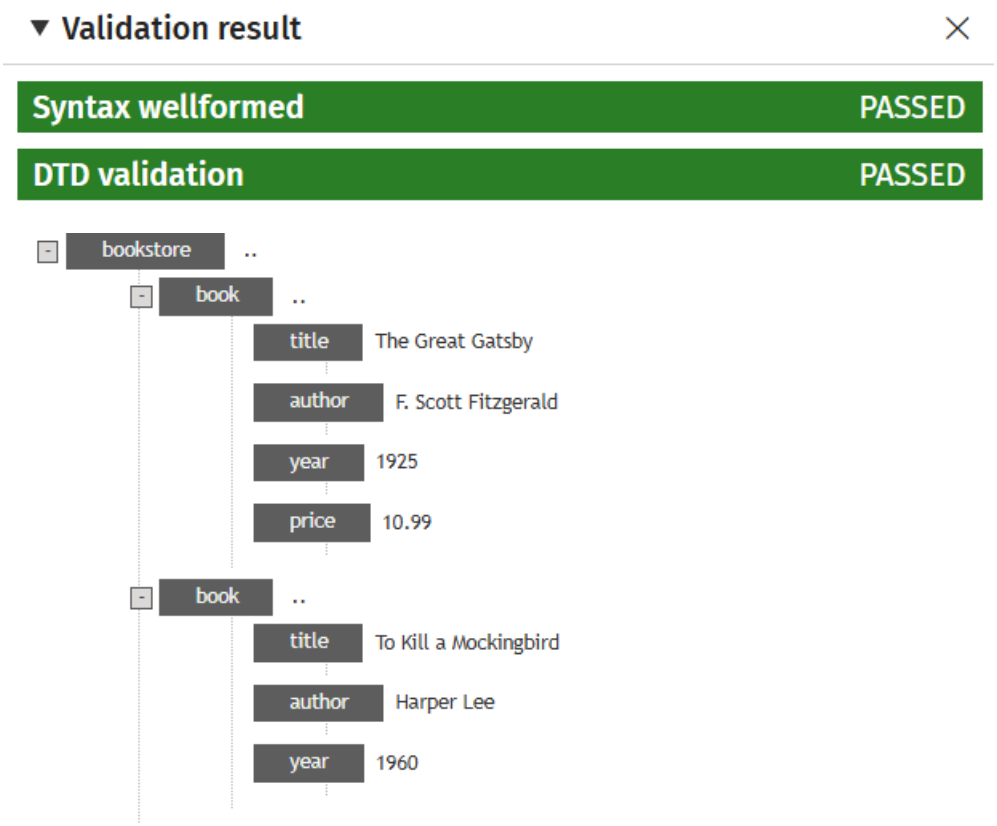


2.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE bookstore [
   <!ELEMENT bookstore (book+)>
   <!ELEMENT book (title, author, year, price?)>
   <!ELEMENT title (#PCDATA)>
   <!ELEMENT author (#PCDATA)>
   <!ELEMENT year (#PCDATA)>
   <!ELEMENT price (#PCDATA)>
]>
<bookstore>
   <book>
      <title>The Great Gatsby</title>
      <author>F. Scott Fitzgerald</author>
      <year>1925</year>
      <price>10.99</price>
   </book>
   <book>
```

```
    <title>To Kill a Mockingbird</title>
    <author>Harper Lee</author>
    <year>1960</year>
    <!-- Price is optional -->
  </book>
</bookstore>
```

OUTPUT

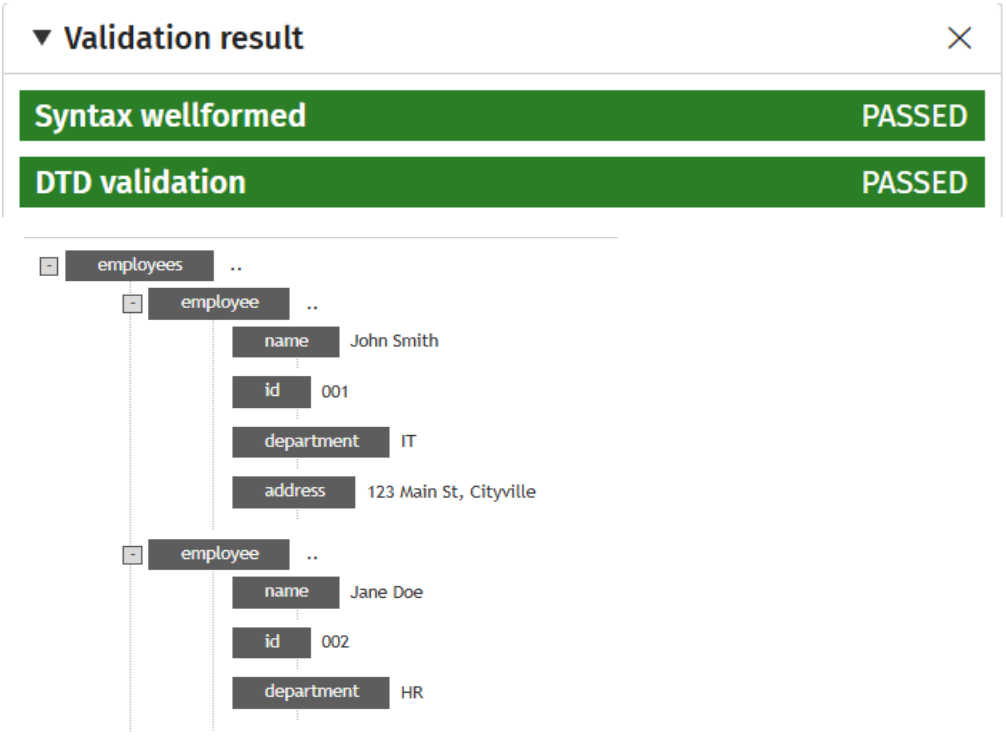


3.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE employees [
  <!ELEMENT employees (employee+)>
  <!ELEMENT employee (name, id, department, address?)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT id (#PCDATA)>
  <!ELEMENT department (#PCDATA)>
  <!ELEMENT address (#PCDATA)>
]>
<employees>
```

```
    <employee>
        <name>John Smith</name>
        <id>001</id>
        <department>IT</department>
        <address>123 Main St, Cityville</address>
    </employee>
    <employee>
        <name>Jane Doe</name>
        <id>002</id>
        <department>HR</department>
        <!-- No address provided -->
    </employee>
</employees>
```

OUTPUT

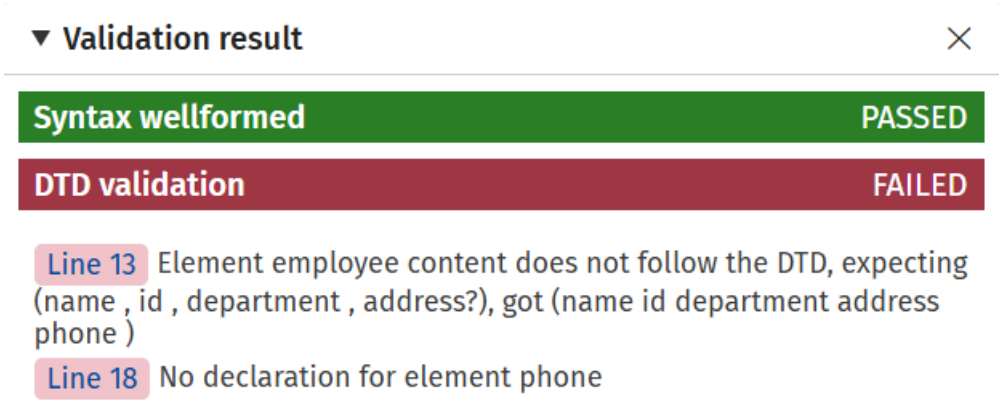


A. <?xml version="1.0" encoding="UTF-8"?>

```
<!DOCTYPE employees [
        <!ELEMENT employees (employee+)>
        <!ELEMENT employee (name, id, department, address?)>
        <!ELEMENT name (#PCDATA)>
```

```
        <!ELEMENT id (#PCDATA)>
        <!ELEMENT department (#PCDATA)>
        <!ELEMENT address (#PCDATA)>
    ]>
    <employees>
        <employee>
            <name>John Smith</name>
            <id>001</id>
            <department>IT</department>
            <address>123 Main St, Cityville</address>
            <phone>90568345128</phone>
        </employee>
        <employee>
            <name>Jane Doe</name>
            <id>002</id>
            <department>HR</department>
            <!-- No address provided -->
        </employee>
    </employees>
```

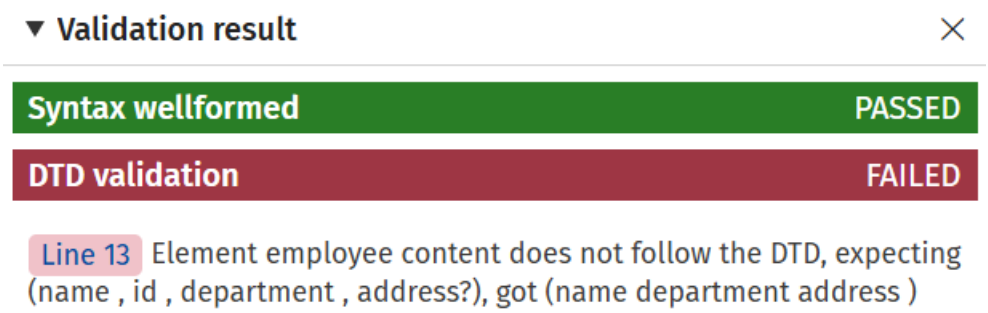
OUTPUT




```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE employees [
    <!ELEMENT employees (employee+)>
    <!ELEMENT employee (name, id, department, address?)>
    <!ELEMENT name (#PCDATA)>
```

```
    <!ELEMENT id (#PCDATA)>
    <!ELEMENT department (#PCDATA)>
     <!ELEMENT address (#PCDATA)>
  ]>
  <employees>
      <employee>
          <name>John Smith</name>
          <!—id not provided -->
          <department>IT</department>
          <address>123 Main St, Cityville</address>
      </employee>
    <employee>
      <name>Jane Doe</name>
      <id>002</id>
      <department>HR</department>
      <!-- No address provided -->
    </employee>
</employees>
```

OUTPUT



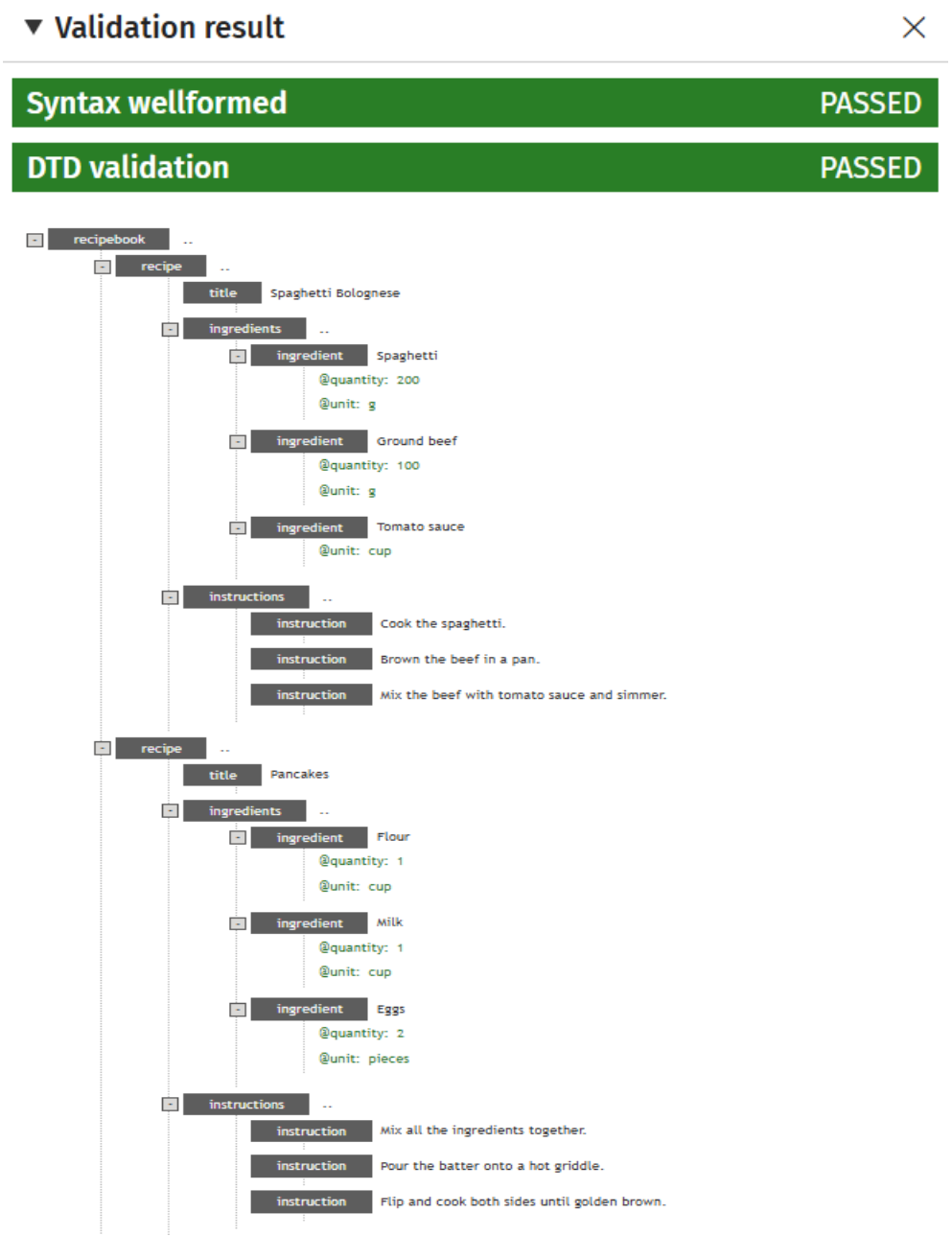
4.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE recipebook [
   <!ELEMENT recipebook (recipe+)>
```

```
    <!ELEMENT recipe (title, ingredients, instructions)>
    <!ELEMENT title (#PCDATA)>
    <!ELEMENT ingredients (ingredient+)>
    <!ELEMENT ingredient (#PCDATA)>
    <!ATTLIST ingredient quantity CDATA #IMPLIED
                         unit CDATA #IMPLIED>
    <!ELEMENT instructions (instruction+)>
    <!ELEMENT instruction (#PCDATA)>
]>
<recipebook>
    <recipe>
        <title>Spaghetti Bolognese</title>
        <ingredients>
            <ingredient quantity="200" unit="g">Spaghetti</ingredient>
            <ingredient quantity="100" unit="g">Ground beef</ingredient>
            <ingredient unit="cup">Tomato sauce</ingredient>
        </ingredients>
        <instructions>
            <instruction>Cook the spaghetti.</instruction>
            <instruction>Brown the beef in a pan.</instruction>
            <instruction>Mix the beef with tomato sauce and simmer.</instruction>
        </instructions>
    </recipe>
    <recipe>
        <title>Pancakes</title>
        <ingredients>
            <ingredient quantity="1" unit="cup">Flour</ingredient>
            <ingredient quantity="1" unit="cup">Milk</ingredient>
            <ingredient quantity="2" unit="pieces">Eggs</ingredient>
        </ingredients>
```

```
    <instructions>
      <instruction>Mix all the ingredients together.</instruction>
      <instruction>Pour the batter onto a hot griddle.</instruction>
      <instruction>Flip and cook both sides until golden brown.</instruction>
    </instructions>
  </recipe>
</recipebook>
```

OUTPUT

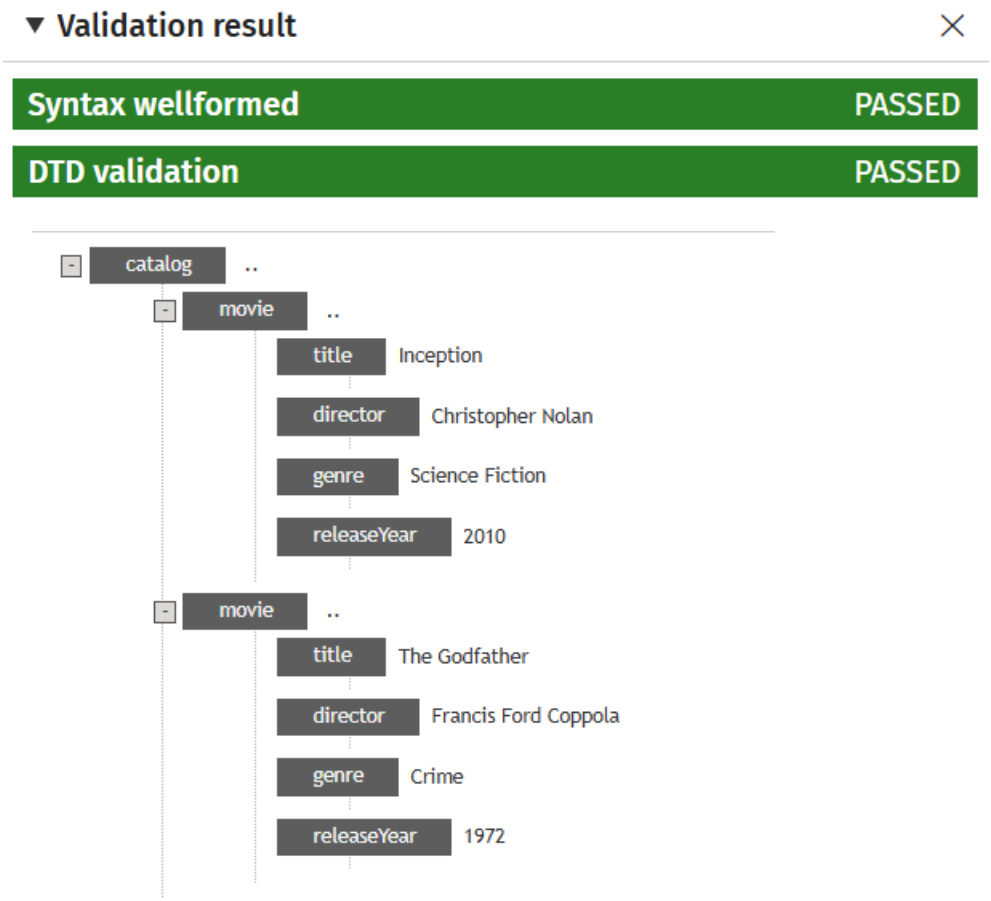


A. By using the **ingredient+ and instruction+** rules in the DTD, it ensures that each recipe must contain **at least one ingredient** and **at least one instruction**. The + sign enforces this by indicating one or more occurrences of the element.

5.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE catalog [
  <!ELEMENT catalog (movie+)>
  <!ELEMENT movie (title, director, genre, releaseYear)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT director (#PCDATA)>
  <!ELEMENT genre (#PCDATA)>
  <!ELEMENT releaseYear (#PCDATA)>
]>
<catalog>
  <movie>
    <title>Inception</title>
    <director>Christopher Nolan</director>
    <genre>Science Fiction</genre>
    <releaseYear>2010</releaseYear>
  </movie>
  <movie>
    <title>The Godfather</title>
    <director>Francis Ford Coppola</director>
    <genre>Crime</genre>
    <releaseYear>1972</releaseYear>
  </movie>
</catalog>
```

OUTPUT



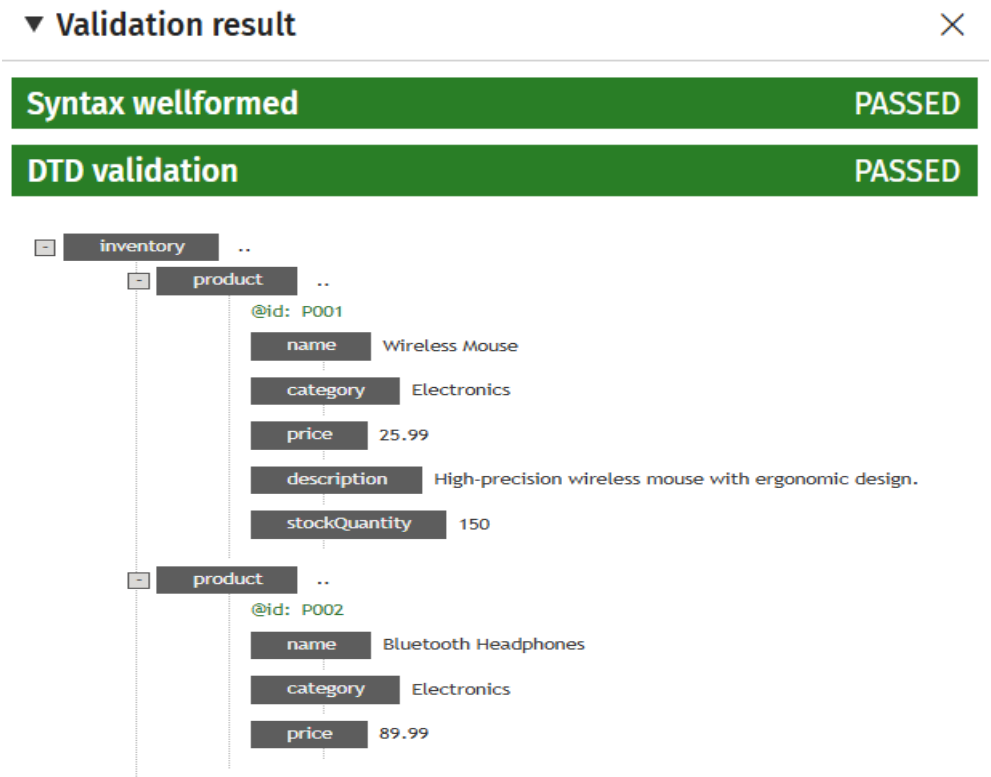
A. DTD can structure and validate XML data, enforcing specific data constraints like a four-digit year is best handled with XML Schema (XSD)

6.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE inventory [
  <!ELEMENT inventory (product+)>
  <!ELEMENT product (name, category, price, description?, stockQuantity?)>
  <!ATTLIST product id CDATA #REQUIRED>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT category (#PCDATA)>
  <!ELEMENT price (#PCDATA)>
  <!ELEMENT description (#PCDATA)>
  <!ELEMENT stockQuantity (#PCDATA)>
]>
```

```
<inventory>
    <product id="P001">
        <name>Wireless Mouse</name>
        <category>Electronics</category>
        <price>25.99</price>
        <description>High-precision wireless mouse with ergonomic design.</description>
        <stockQuantity>150</stockQuantity>
    </product>
    <product id="P002">
        <name>Bluetooth Headphones</name>
        <category>Electronics</category>
        <price>89.99</price>
        <!-- Optional elements can be omitted -->
    </product>
</inventory>
```

OUTPUT



A. The **<!ATTLIST product id CDATA #REQUIRED>** line in the DTD specifies that the id attribute of the product element is required. This means every product element in the XML document must have an id attribute.

7.

inventory.dtd

```
<!ELEMENT product (name, category, price, description?, stockQuantity?)>

<!ATTLIST product id CDATA #REQUIRED>

<!ELEMENT name (#PCDATA)>

<!ELEMENT category (#PCDATA)>

<!ELEMENT price (#PCDATA)>

<!ELEMENT description (#PCDATA)>

<!ELEMENT stockQuantity (#PCDATA)>
```

inventory.xml

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

<!DOCTYPE inventory SYSTEM "inventory.dtd">

<inventory>

  <product id="P001">

    <name>Wireless Mouse</name>

    <category>Electronics</category>

    <price>25.99</price>

    <description>High-precision wireless mouse with ergonomic design.</description>

    <stockQuantity>150</stockQuantity>

  </product>

  <product id="P002">

    <name>Bluetooth Headphones</name>

    <category>Electronics</category>

    <price>89.99</price>

    <!-- Optional elements can be omitted -->

  </product>

</inventory>
```

A. To change from an internal DTD to an external DTD in your XML file, replace the internal DTD declaration with:

<!DOCTYPE parentelement SYSTEM "filename.dtd">

RESULT

Thus , successfully practiced XML along with DTD using the given questions