New Perspectives on XML Comprehensive, 3rd Edition

Additional Project

Tutorial 5

Jenna has been working on creating a DVD library where she can list all of the DVDs that she has. Some she bought herself, others were gifts from her mom, dad, sister, grandpa, and grandma. The root element for her library is dvdlibrary with one or more dvd elements. Each dvd has a title, description, year, company, favorite characters, favorite bonus features, and purchase information. Her characters elements each have one or more character elements. Her bonus elements each have one or more features. Her purchased elements each have price, location, and date. The dvd element has two attributes, bonus element has one attribute, and the purchased element has no attributes.

Jenna wishes to create an XSLT style sheet to use to display the DVD collection information attractively. She has already created a CSS style sheet for the page. Complete the following:

1. Using your text editor, open the file***dvdlibrary.txt***.Review the contents to familiarize yourself with the structure of the XML file.
2. Add a comment in the prolog with a description of the file, your name, date, and the filename.
3. Save the file as ***dvdlibrary.xml***.
4. Add a processing instruction after the comment section that attaches the ***dvdlibrary.xsl*** style sheet to this XML document. Close the file, saving your changes.
5. Using your text editor, open the file ***dvdlibrarytxt.xsl***. Enter your name and the date in the comment section of the file, and save it as ***dvdlibrary.xsl***.
6. Go to the ***dvdlibrary.xsl*** file in your text editor and go down to the root template. Add the following commands to the template:  
     
   <html>

<head>

<title>Jenna's DVD Collection</title>

<link href="dvdlibrarystyles.css"  
rel="stylesheet" type="text/css" />

</head>

<body>

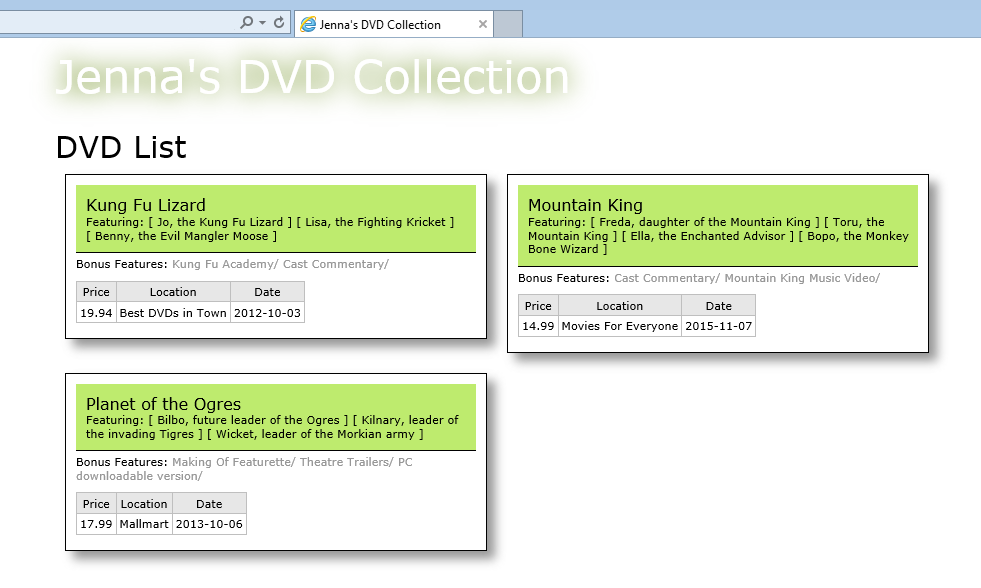
<header>

<h1>Jenna's DVD Collection</h1>

</header>

</body>

</html>

1. Within the root template directly above the </body> tag, insert the following:  
     
   <section>  
   <h1>DVD List</h1>  
   *dvd template*</section>  
     
   where *dvd template* applies the template for the dvdlibrary/dvd path, sorted by the title element.
2. Create a template for the dvd element to display information on each movie. Add the following HTML code to the template:  
     
   <article>  
   </article>
3. Within the <article></article> tags, insert the following to display the title and list of characters for each DVD:  
     
   <h1>*title*</h1>  
   <h2>Featuring: *[character] [character] …*</h2>  
     
   where *title* is the value of the title element and *[character] [character] …* is the list of characters in the movie. (Hint: Use the for-each instruction to go through each character element in the characters/character path.)
4. Below the h2 heading, insert the following:  
     
   <p>  
    Bonus Features:  
   <span>  
   *feature/feature/ …*  
   </span>  
   </p>  
     
   where *feature/feature/ …* is a list of the feature values within the bonus/feature path separated with the “/” symbol. (Hint: Use the for-each instruction with the bonus/feature path and display the value of the context node within the for-each element.)
5. Below the closing </p> tag, insert the following table:  
     
   <table>  
   <tr>  
   <th>Price</th>  
   <th>Location</th>  
   <th>Date</th>  
   </tr>  
   <tr>  
   <td>*price*</td>  
   <td>*location*</td>  
   <td>*date*</td>  
   </tr>  
   </table>  
     
   where *price*, *location*, and *date* are the values of the price, location, and date elements.
6. Save your changes to the file.
7. Generate a result document using either an XML editor or your Web browser. Verify that the layout and content of each item matches that shown in the figure below:  
   
8. Submit the completed files to your instructor.