New Perspectives on XML Comprehensive, 3rd Edition

Additional Project

Tutorial 8

Maria Soaresworks at the Denison Public Library. She has been developing XSLT applications to display and make use of the library's store of XML data. The library director has asked her to develop an application that the librarians can use to better assess the status of overdue items.

Maria would like to generate a report listing items currently checked out from the library. Items are overdue when they have not been returned by the due date. Items that are more than 30 days overdue are classified as "Long Overdue", and require a phone call to the patron who checked out the item. Items that are more than 90 days overdue are classified as "Lost", and require suspension of the patron's library card until the library is reimbursed for the cost of the item.

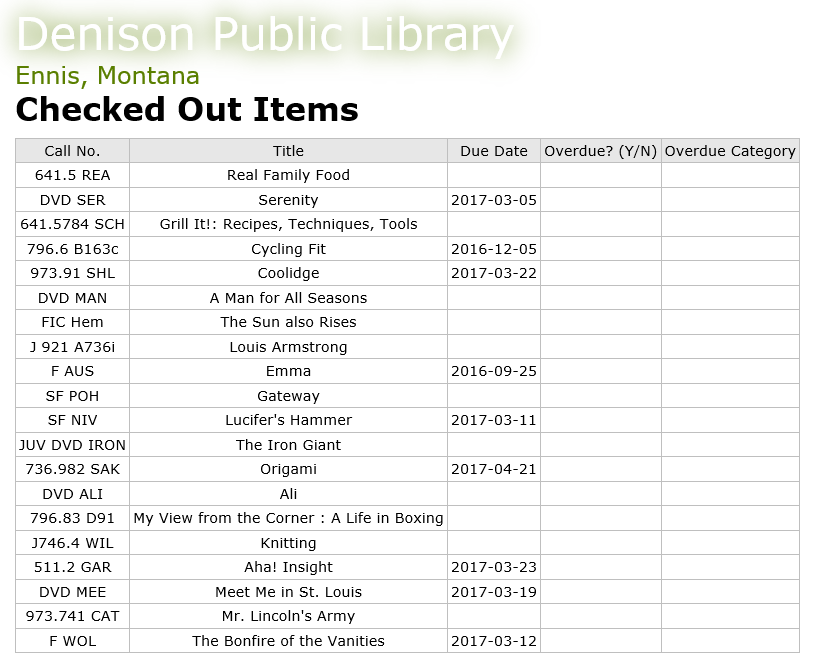
Maria would like the new report to list all library items. Checked out items should display their due date, an indicator whether they are overdue and, if they are overdue, display a label indicating which overdue category they are in.

Complete the following:

1. Using your text editor, open the file ***itemstxt.xml***. Review the contents to familiarize yourself with the structure of the XML file.
2. Enter your name and the date in the comment section at the top of the file, and save the file as ***items.xml***.
3. Add a processing instruction after the comment section that attaches the library.xsl style sheet to this XML document. Close the file, saving your changes.
4. Using your text editor, open the file ***librarytxt.xsl***.
5. Enter your name and the date in the comment section of the file, and save it as ***library.xsl***.
6. In the opening stylesheet tag below the comment section, change the version from "1.0" to "2.0".Save your changes to the file.
7. If you are using Saxon in Java command line mode, go to the folder containing your data files, and run the following command to generate the checkouts.html result document:

javanet.sf.saxon.Transform items.xml library.xsl   
-o:checkouts.html

Otherwise, use the commands appropriate to your XSLT 2.0 processor to run the transformation.

1. Use your web browser to open the ***checkouts.html*** file. The figure below shows the current appearance of the report.  
     
   
2. The first feature Maria would like to add is to display the date above the table. The date April 12, 2017 will be used for testing the application. Later, Maria will replace this hard coded date with a variable representing the current date.  
     
   Below the xsl:output tag, insert code to create a global variable named *thisDate* with the data type xs:date, containing the value '2017-04-12'.
3. Within the root template, above the *Checked Out Items*<h1> tag, insert an <h2> tag containing a value-of element to display the value of the thisDate variable. Use a date picture to format the date as "April 12, 2017".
4. The Overdue column of the report should contain 'Y' if the item is overdue (the due date is before the date the report is run) or 'N' if it is not overdue. Within the "overdueCell" <td> tag, declare a variable named *overdue*. Use a conditional if statement to test whether the value of the thisDate value is greater than the value of the return attribute of the status element. If so, set the variable's value to 'Y', else set the value to 'N'. On the next line, insert a value-of element to display the value of the overdue variable.
5. Save your changes to the style sheet and then regenerate the result document into the ***checkouts.html*** file using your XSLT 2.0 processor.
6. View the ***checkouts.html*** file in your web browser. Confirm that items are labeled appropriately in the Overdue column.
7. The Overdue Category column should contain "Long Overdue" if the item's due date is more than 30 days before the date the report is run. The column should contain "Lost" if the item's due date is more than 90 days before the date the report is run.  
     
   Within the categoryCell<td> tag, declare two variables of data type date: *longoverdueDate* and *lostDate*. The longoverdueDate variable should be set to a value 30 days earlier than the thisDate variable. (Hint: Subtract a dayTimeDuration element of 30 days from the thisDate variable.) The lostDate variable should be set to a value 90 days earlier than the thisDate variable.
8. Next, declare a variable named *category*. Use an else if conditional statement to determine the category value to display in the cell. If the return attribute of the status element is less than the value of the lostDate variable, set the category variable to 'Lost'. Else, if the return attribute is less than the longoverdueDate, set the category variable to 'Long Overdue'. Else, if the value of the thisDate variable is less than the return attribute, set the category variable to 'Overdue'. Else, set the category variable to an empty string, ''.  
     
   On the next line, insert a value-of element to display the value of the category variable.
9. Save your changes to the style sheet and then regenerate the result document into the ***checkouts.html*** file using your XSLT 2.0 processor.
10. View the ***checkouts.html*** file in your web browser. Verify that the layout of the web page matches that shown in the figure below:  
      
    
11. Submit the completed files to your instructor.