Homework8 Zhou, Tiancong

the CSV input file

date, ipaddress, userAgent

2013-01-02,12.22.333.44, http://www.zorro.com

2013-01-03,123.33.44.55,Mozilla/5.0 (Windows NT 6.1; rv:27.0) Gecko/20100101 Firefox/27.0

2013-01-03,123.33.44.55,Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)

2013-01-06,256.19.44.55,Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; .NET CLR 2.0.50727)

2013-01-07,123.33.44.55,Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)

2013-01-06,123.33.44.55,Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)

2013-01-06,212.38.44.55,Mozilla/3.0 (x86 [en] Windows NT 5.1; Sun)

2013-01-02,212.38.44.55,Mozilla/3.0 (x86 [en] Windows NT 5.1; Sun)

2013-02-02,5.255.23.25,Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)

2013-02-05,5.255.23.25,Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)

• the XML generated by the PHP

<?xml version="1.0" encoding="UTF-8"?>  
<rows>  
 <row>  
 <date>2013-01-02</date>  
 <ipaddress>12.22.333.44</ipaddress>  
 <userAgent>http://www.zorro.com</userAgent>  
 </row>  
 <row>  
 <date>2013-01-03</date>  
 <ipaddress>123.33.44.55</ipaddress>  
 <userAgent>Mozilla/5.0 (Windows NT 6.1; rv:27.0) Gecko/20100101 Firefox/27.0</userAgent>  
 </row>  
 <row>  
 <date>2013-01-03</date>  
 <ipaddress>123.33.44.55</ipaddress>  
 <userAgent>Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)</userAgent>  
 </row>  
 <row>  
 <date>2013-01-06</date>  
 <ipaddress>256.19.44.55</ipaddress>  
 <userAgent>Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; .NET CLR 2.0.50727)</userAgent>  
 </row>  
 <row>  
 <date>2013-01-07</date>  
 <ipaddress>123.33.44.55</ipaddress>  
 <userAgent>Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)</userAgent>  
 </row>  
 <row>  
 <date>2013-01-06</date>  
 <ipaddress>123.33.44.55</ipaddress>  
 <userAgent>Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)</userAgent>  
 </row>  
 <row>  
 <date>2013-01-06</date>  
 <ipaddress>212.38.44.55</ipaddress>  
 <userAgent>Mozilla/3.0 (x86 [en] Windows NT 5.1; Sun)</userAgent>  
 </row>  
 <row>  
 <date>2013-01-02</date>  
 <ipaddress>212.38.44.55</ipaddress>  
 <userAgent>Mozilla/3.0 (x86 [en] Windows NT 5.1; Sun)</userAgent>  
 </row>  
 <row>  
 <date>2013-02-02</date>  
 <ipaddress>5.255.23.25</ipaddress>  
 <userAgent>Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)</userAgent>  
 </row>  
 <row>  
 <date>2013-02-05</date>  
 <ipaddress>5.255.23.25</ipaddress>  
 <userAgent>Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)</userAgent>  
 </row>  
</rows>

• the PHP program

<?php

ini\_set("display\_errors", true);

$inputFilename = 'hw8.csv';

$outputFilename = 'output.xml';

// Open csv to read

$inputFile = fopen($inputFilename, 'rt');

// Get the headers of the file

$headers = fgetcsv($inputFile);

// Create a new dom document with pretty formatting

$doc = new DomDocument();

$doc->formatOutput = true;

// Add a root node to the document

$root = $doc->createElement('rows');

$root = $doc->appendChild($root);

// Loop through each row creating a <row> node with the correct data

while (($row = fgetcsv($inputFile)) !== FALSE)

{

$container = $doc->createElement('row');

// $i is the array index (0,1,2,..) and $header is the value at $i

foreach ($headers as $i => $header)

{

$child = $doc->createElement($header);

$child = $container->appendChild($child);

$value = $doc->createTextNode($row[$i]);

$value = $child->appendChild($value);

}

$root->appendChild($container);

}

// here we are sending the XML back to the client

// However, realize that this is a string of XML that XSLT can operate on

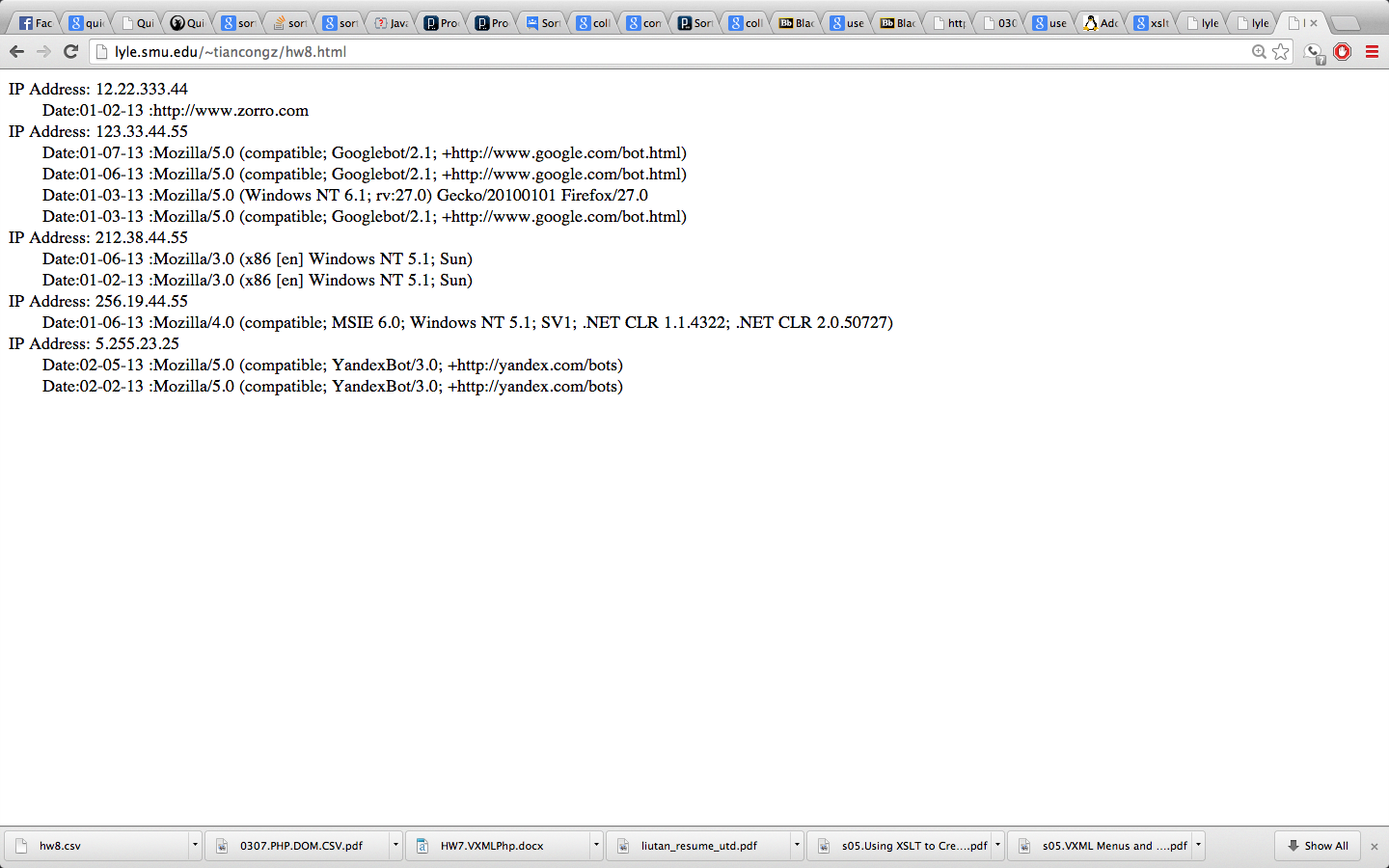
echo $doc->saveXML();

?>

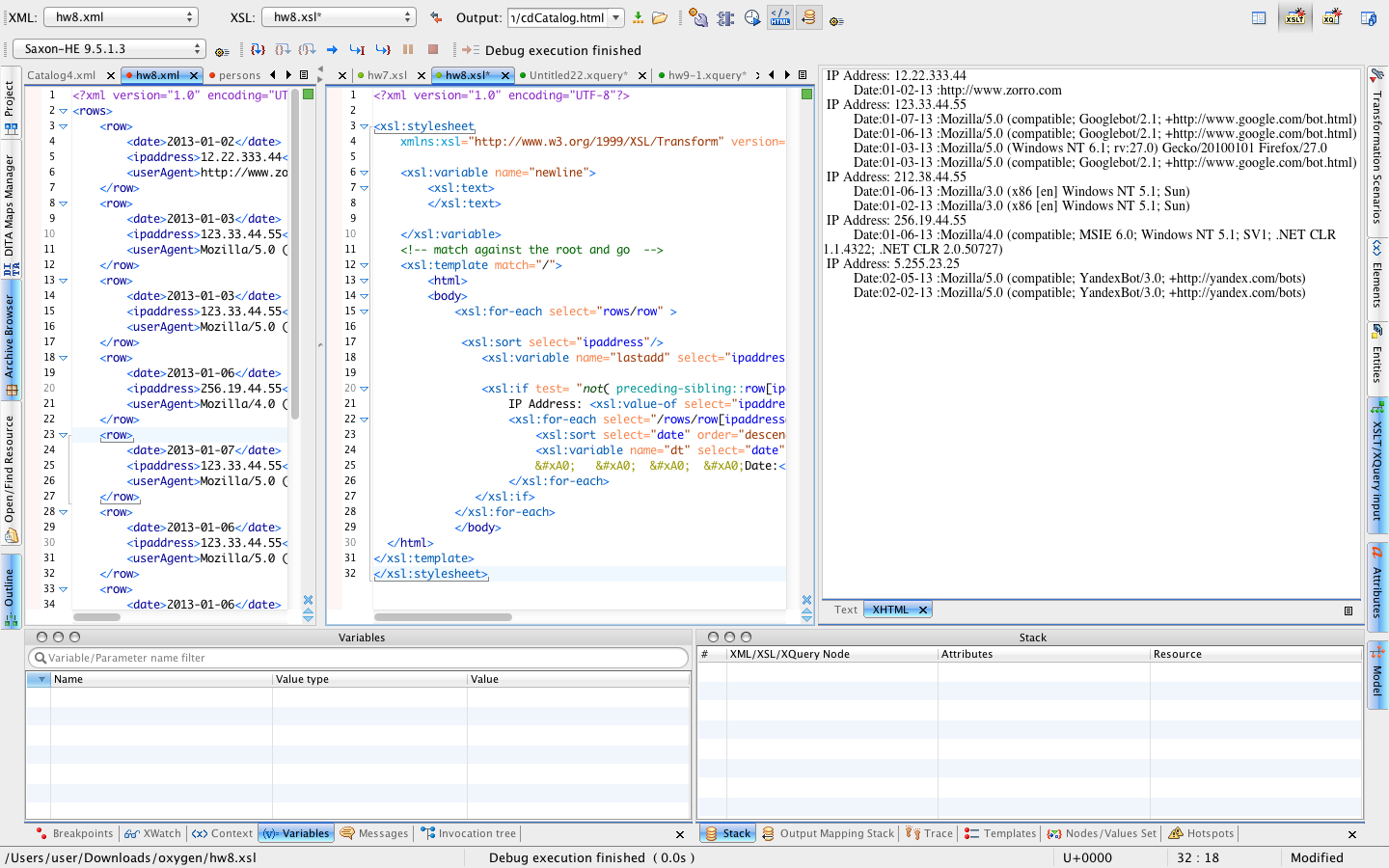
• xslt program:

<?xml version="1.0" encoding="UTF-8"?>  
  
<xsl:stylesheet  
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">  
   
 <xsl:variable name="newline">  
 <xsl:text>  
 </xsl:text>  
   
 </xsl:variable>  
 <!-- match against the root and go -->  
 <xsl:template match="/">  
 <html>  
 <body>  
 <xsl:for-each select="rows/row" >  
   
 <xsl:sort select="ipaddress"/>   
 <xsl:variable name="lastadd" select="ipaddress"></xsl:variable>  
   
 <xsl:if test= "not( preceding-sibling::row[ipaddress=$lastadd])">  
 IP Address: <xsl:value-of select="ipaddress"/> <br/>  
 <xsl:for-each select="/rows/row[ipaddress=$lastadd]">  
 <xsl:sort select="date" order="descending"></xsl:sort>  
 <xsl:variable name="dt" select="date" ></xsl:variable>  
 &#xA0; &#xA0; &#xA0; &#xA0;Date:<xsl:value-of select="format-date($dt,'[M01]-[D01]-[Y01]')"/>&#xA0;:<xsl:value-of select="userAgent"/><br/>  
 </xsl:for-each>  
 </xsl:if>  
 </xsl:for-each>  
 </body>  
 </html>  
</xsl:template>  
</xsl:stylesheet>

• a screen shot of the web output



• a link to the php program that does the work and displays the HTML



link to the html file: <http://lyle.smu.edu/~tiancongz/hw8.html>