UNIVERSITY OF WESTMINSTER

5COSC020W DATABASE SYSTEMS

2022-2023 Tutorial 08 & 09

Navigating through & rendering XML docs – XPath and XSLT

Model Answer

Case Study

Westmuni is a university specialised in the delivery of wide range of IT courses to Undergraduate students.

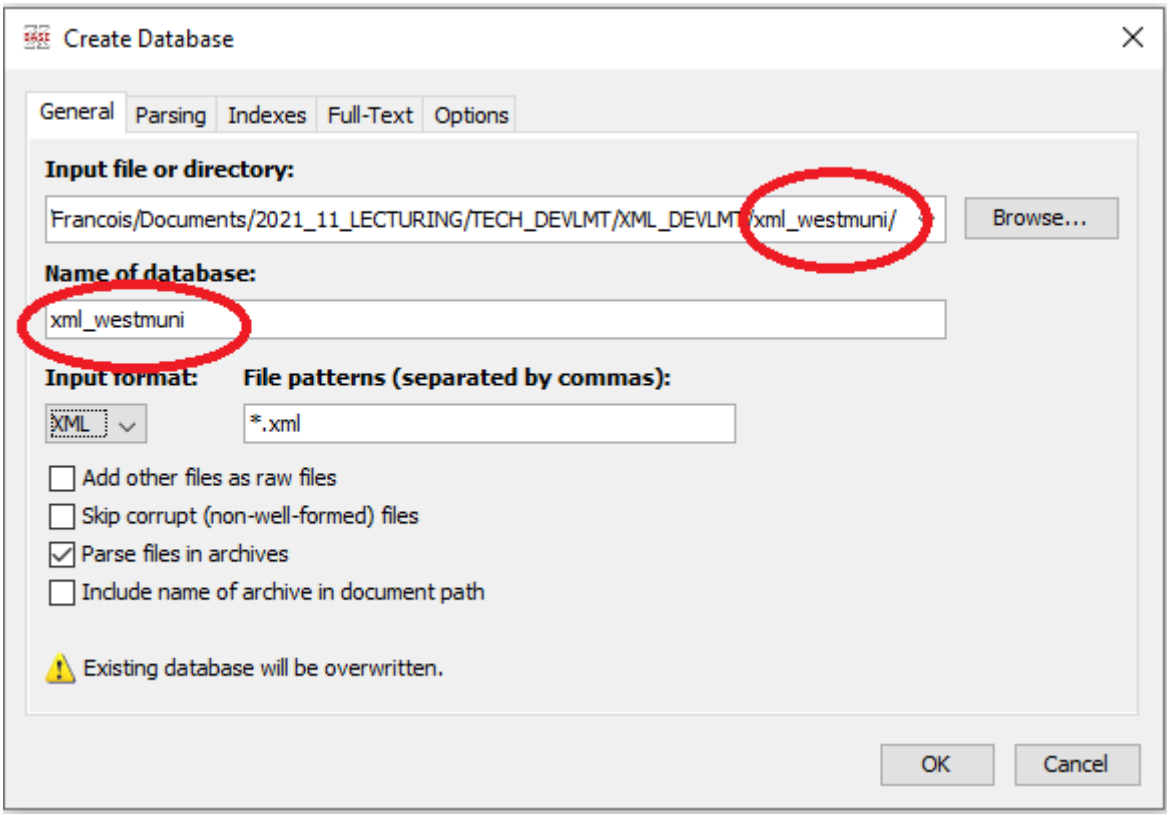
Setting Up: Creating an XML database on BaseX

i. Access the XML document from Blackboard

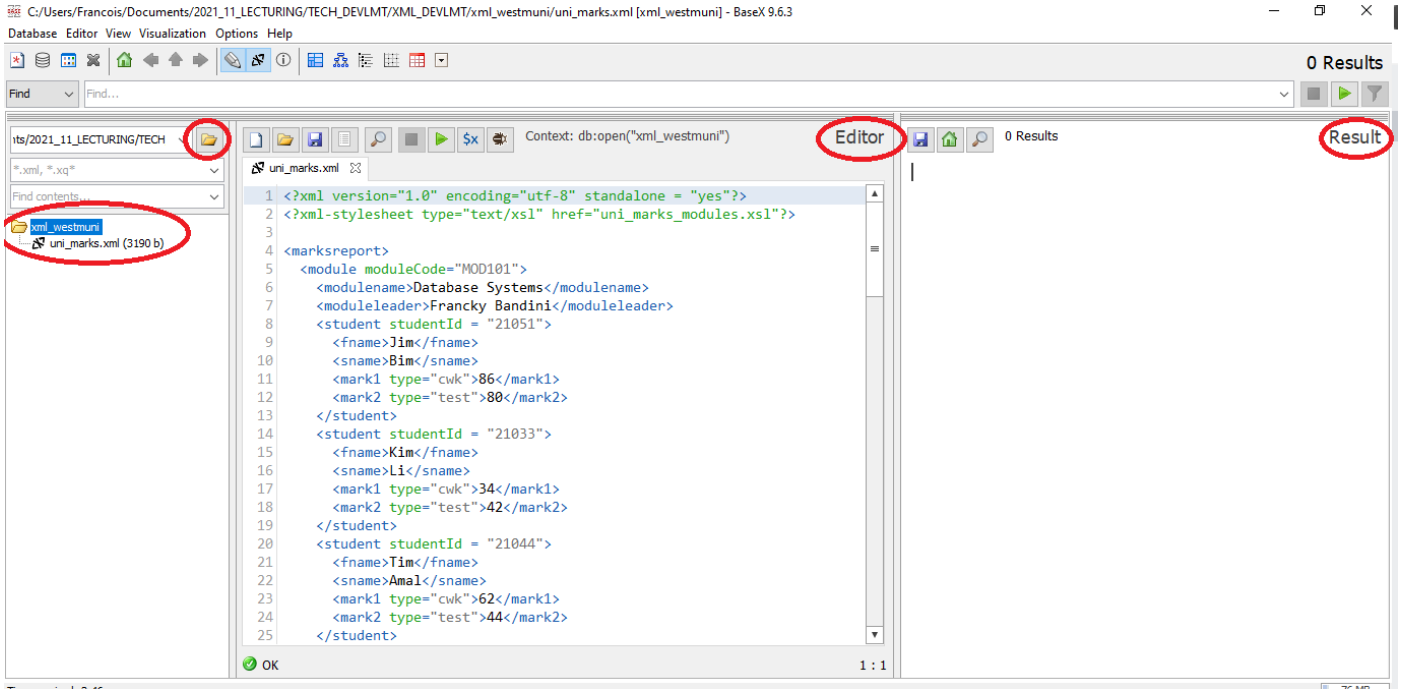
1. Create a **directory** locally on your machine called **xml_westmuni**.
2. Get the **XML document** called **uni_marks.xml** from Blackboard under 'Learning Resources and 'Section 3 – XML'. Do not click on the XML file but instead right-click on it and select "save-link as".
3. Save the uni_marks.xml document in your **xml_westmuni directory**.

ii. Create an XML database on BaseX

1. Locate **BaseX** on AppsAnywhere https://appsanywhere.westminster.ac.uk and launch it.
2. Alternatively, download BaseX from https://basex.org/ if you are using your own machine
3. Create a **New Database** on BaseX. Click on "Database" on the top nav bar and select "New".
4. Click browse and locate your xml_westmuni directory.
5. Name your **XML database** as **xml_westmuni** (same name as your directory) and click OK.



6. Use the browsing tool on the left hand-side to bring up your **xml_westmuni directory** and open the **uni_marks.xml** document in the editor. You can also use the View tab on the menu on the top nav bar to display the result pane on the right hand-side and hide any other panes, if you so wish.



TUTORIAL 08 & 09 PART 1: XPath

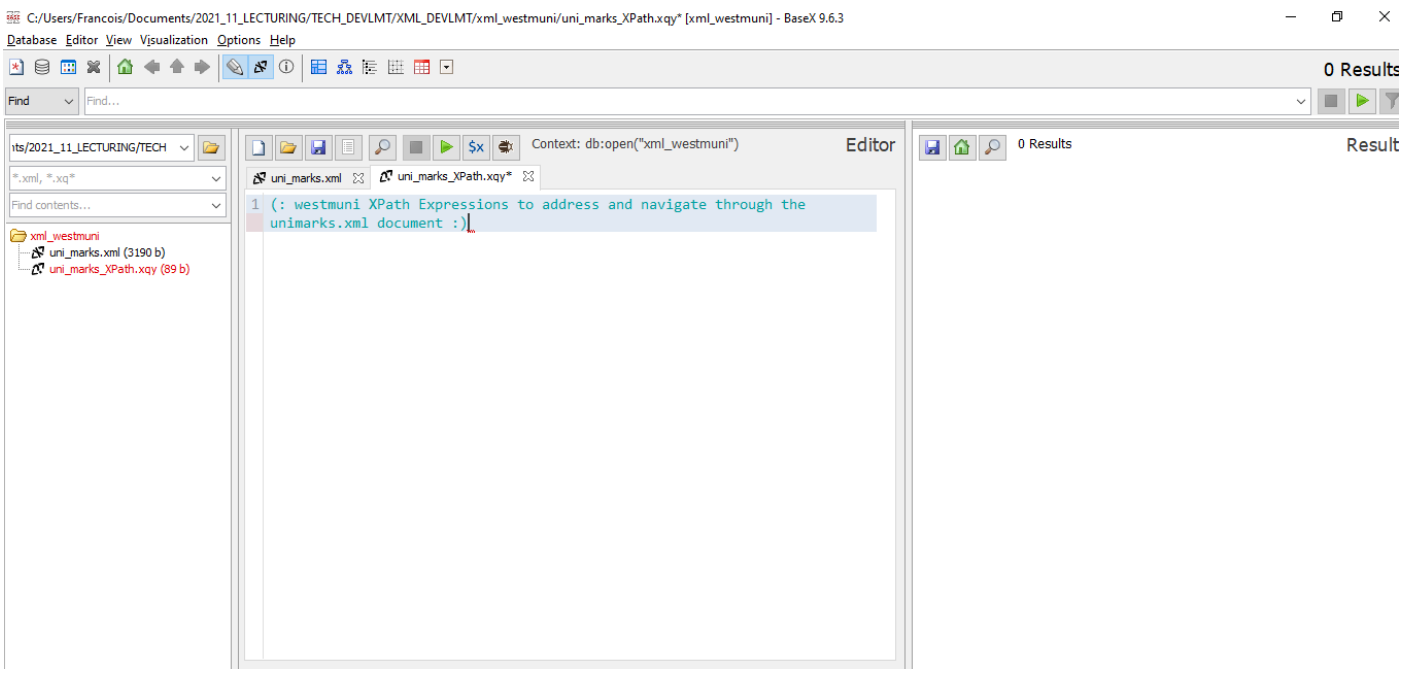
Tutorial 08 & 09 Task 01: Creating your XPath file on BaseX

1. Create a new file (of type .xqy) to write your XPath expressions. Click on the "New" icon to open a new tab.

2. Click on the "Save" icon to save the new file as **uni_marks_XPath.xqy** in the same directory xml_westmuni.

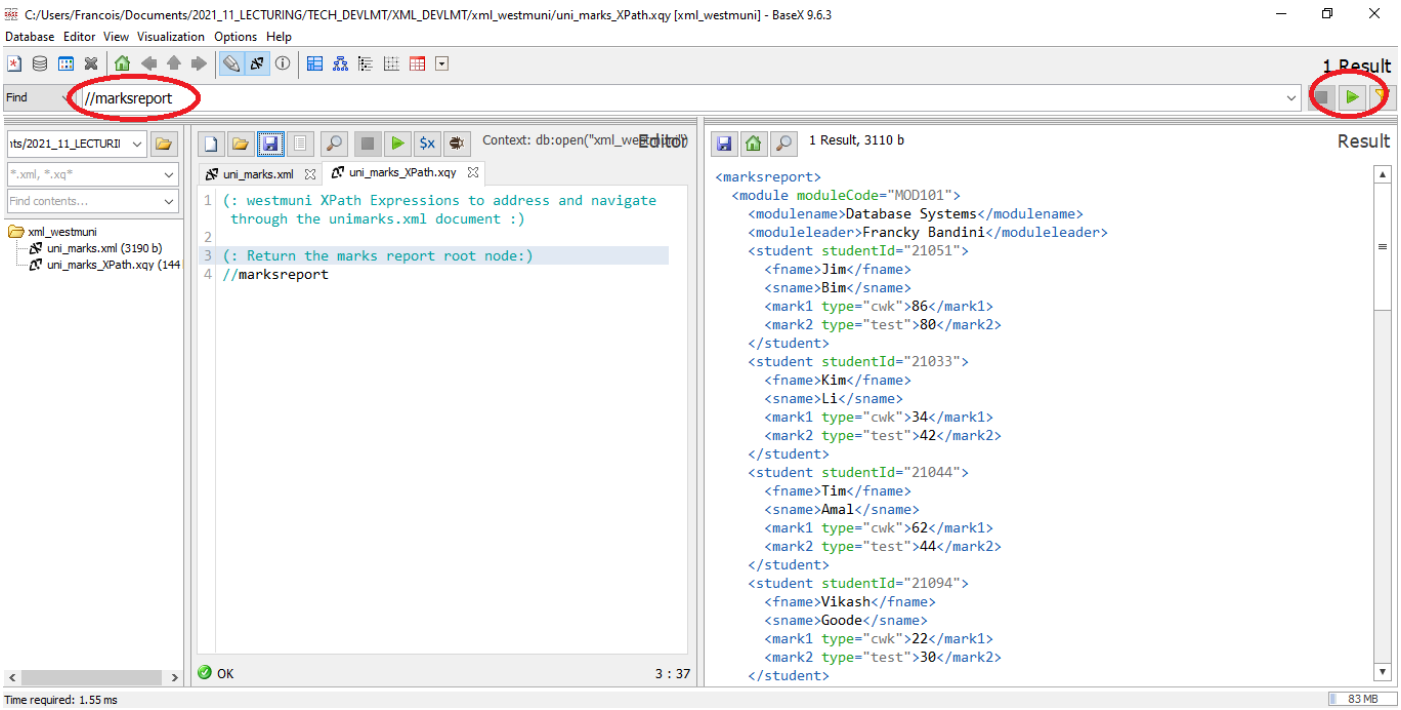
3. Type in a comment between (: and :) at the top of your **uni_marks_XPath.xqy** file e.g.

```
(: westmuni XPath Expressions to address and navigate through the unimarks.xml document :)
```



4. Write your first XPath expression in your editor to return the root note in your editor `//marksreport`

5. Copy and paste it in the box above the editor and click on the green “run query” icon on the right hand-side to execute it. View the result on the pane on the right-hand side.



7. Continue writing your XPath expressions in the editor. For every expression, copy and paste it in the above box to run it and view the output.

Tutorial 08 & 09 Question 01: Simple XPath expressions

a) Return the modules nodes and descendants (using an absolute path).

```
/marksreport/module
```

b) Return the modules nodes and descendants (using a relative path).

```
//module
```

c) Return the module names and module leaders (using an absolute path).

```
/marksreport/module/modulename | /marksreport/module/moduleleader
```

d) Return the module names and module leaders and descendants (using a relative path).

```
//modulename | // moduleleader
```

Tutorial 08 & 09 Question 02: XPath expressions with conditions

a) Return the module details for the module called "Database Systems".

```
//module[modulename="Database Systems"]
```

b) Return the surnames of the students on the "Database Systems" module.

1st answer

```
//module[modulename="Database Systems"]/student/sname
```

2nd answer

```
//module[modulename="Database Systems"]//sname
```

c) Return the details of the students who have scored more than 40 in the first component.

```
//module/student[mark1>=40]
```

d) Return the details of the students on the Database Systems module who have scored more than 40 in the first component of the assessment.

```
//module[modulename="Database Systems"]/student[mark1>=40]
```

e) Return the students on the module identified by the code "MOD102".

```
//module [@moduleCode="MOD102"]/student
```

Tutorial 08 & 09 Question 03: XPath expressions using wildcards

a) Return any node for which the first name matches Jim. Use a wildcard.

```
//*[fname="Jim"]
```

b) Return the names of all the modules in which Vikash is enrolled. Use a wildcard.

```
//*[fname="Vikash"]/../modulename
```

c) Return all the marks scored by the student Vikash. Use a wildcard.

```
//*[fname="Vikash"]/./mark1 | //*[fname="Vikash"]/./mark2
```

d) Return the surname of the students who scored exactly 86 in the first component.

```
//*[mark1=86]/./sname
```

e) Return the students' surnames in the module identified by the code MOD103. Use a wildcard.

//*[@moduleCode="MOD103"]//sname

Tutorial 08 & 09 Question 04: XPath expressions using logical operators

a) Return the details of modules that have 102 and 103 as module codes using the or logical operator.

```
//module[(@moduleCode="MOD102") or (@moduleCode="MOD103")]
```

b) Return the details of modules that do not have the module code MOD102 using the not logical operator.

```
//module[not(@moduleCode="MOD102")]
```

c) Return the details of students who for mark1 have scored a mark between 50 and 60 (inclusive) using the and logical operator.

```
//student[mark1>=50 and mark1<=60]
```

d) Return the details of students that have scored 76 in a test for the first component (mark 1).

```
//student[(mark1[@type="test"]) and (mark1 =76)]
```

e) Return the details of the students that have either scored 75 and over in a test for the first component (mark1) or 45 an over in an exam for the second component (mark 2).

```
//student[((mark1[@type="test"]) and (mark1>=75)) or ((mark2[@type="exam"]) and (mark2
>=45))]
```

TUTORIAL 08 & 09 PART 2: XSLT

Tutorial 08 & 09 Task 02: Use an XSLT file to render the XML document and view a list of modules with the students and marks

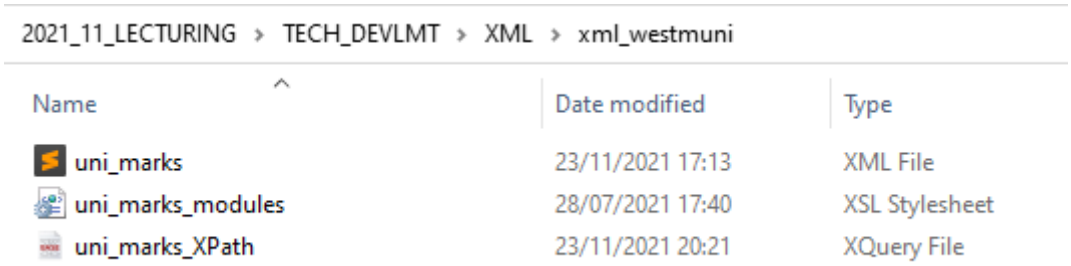
i. Access the XSLT file from Blackboard

1. Get the **XSLT file** called **uni_marks_modules.xsl** from Blackboard under 'Learning Resources and 'Section 3 – XML'. Do not click on the XSLT file but instead right-click on it and select "save-link as".

2. Save the **uni_marks_modules.xsl** file in your **xml_westmuni directory**.

3. Check that your **xml_westmuni directory** now contains the following files

   - The XML document **uni_marks.xml**
   - The XPath file **uni_marks_XPath.xqy**
   - The XSLT file **uni_marks_modules.xsl**



ii. Modify the XML file to call the XSLT Stylesheet

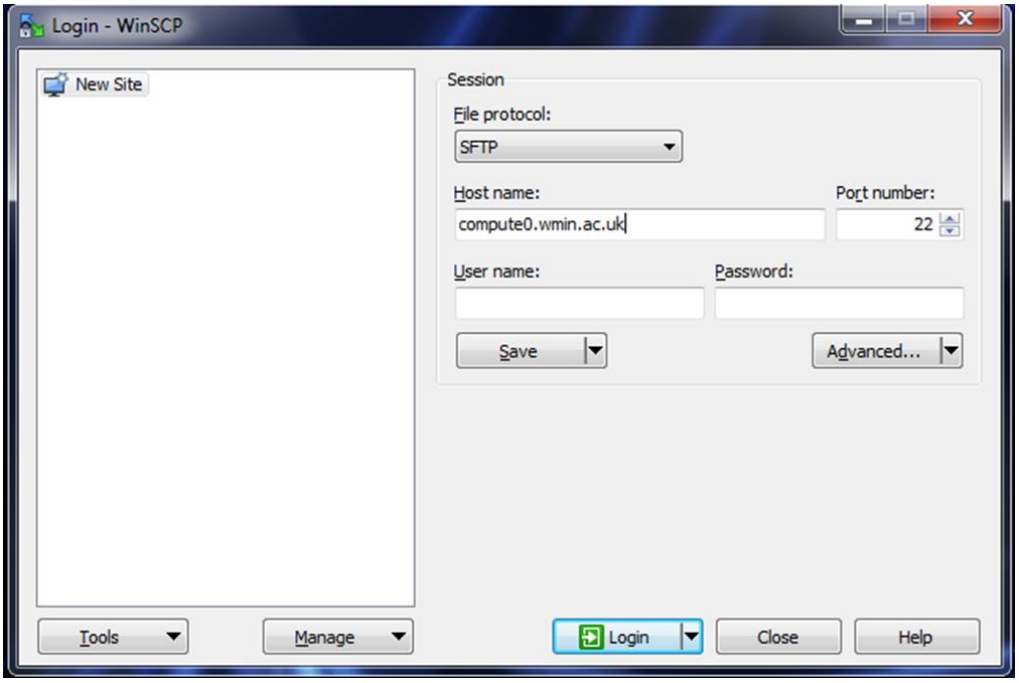
Edit your **XML document** called **uni_marks.xml** and add this line (on line 2) as a reference to the XSLT stylesheet.

```
<?xml-stylesheet type="text/xsl" href="uni_marks_modules.xsl"?>
```

iii. Upload the xml_westmuni directory onto the server

1. Go to AppsAnywhere on https://appsanywhere.westminster.ac.uk and launch **WinSCP** if you are running Windows, or **FileZilla** if you are running macOS.

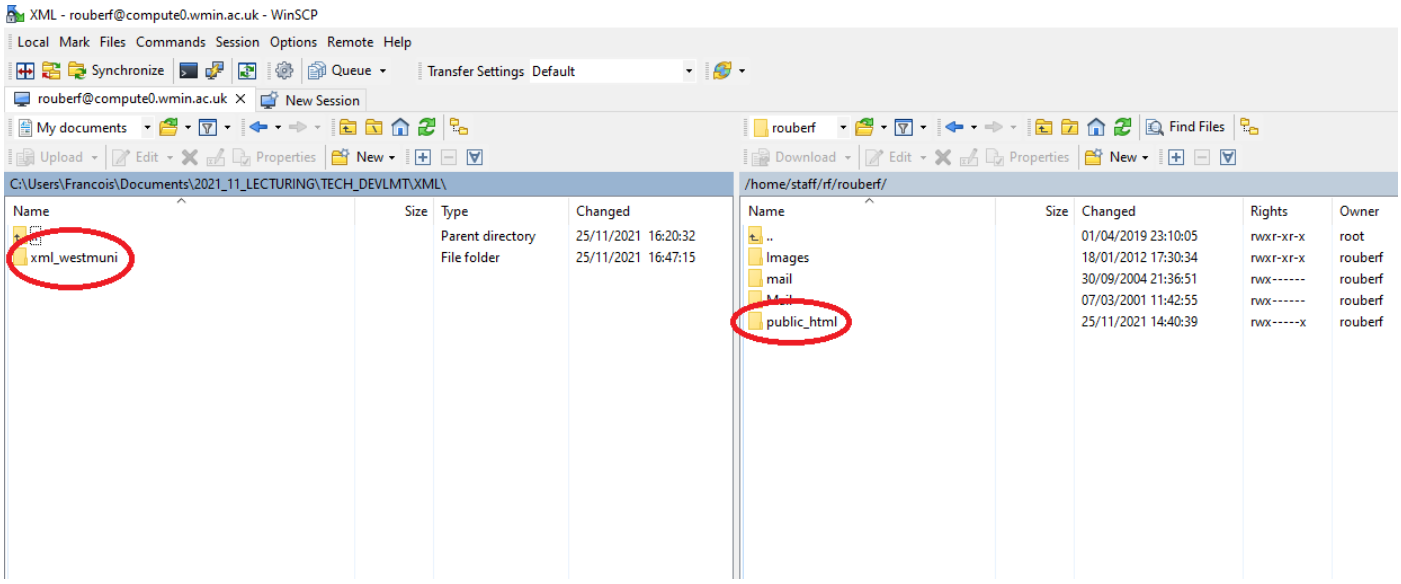
2. With **WinSCP**, fill in your details and click Login.



- **The Host name**: compute0.wmin.ac.uk
- **Your Uni login name**: w + 7 digits of your id number with w in lower case e.g. w1234567
- **Your Uni password**

Make sure that you can see a **public_html directory** on the right pane.

3. Upload your **local xml_westmuni directory** (left pane) by dragging the whole directory across **INSIDE** the **remote public_html directory** on the server (right pane).



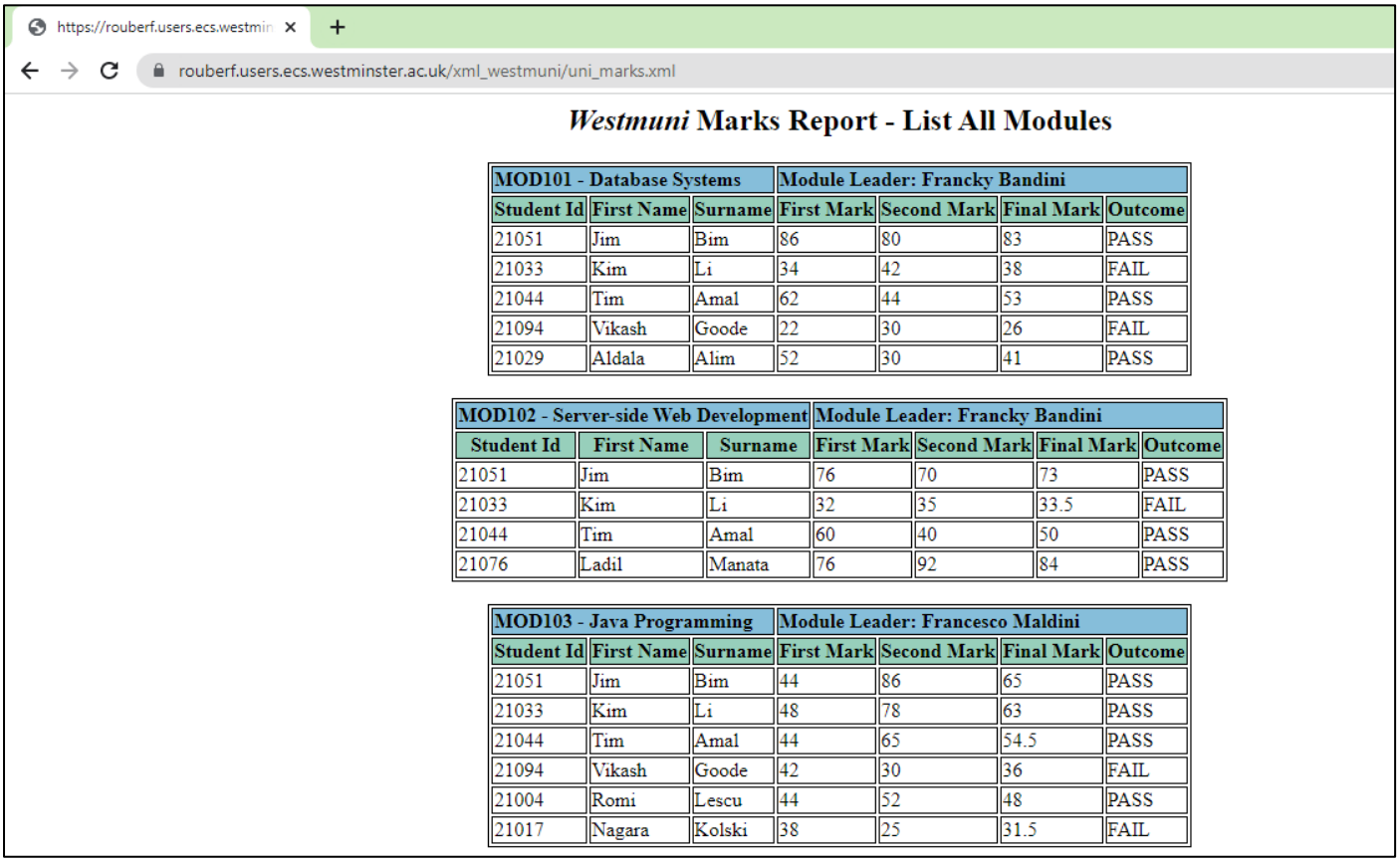
iv. View the XML document as rendered by the XSLT file

1. Open a browser and enter the following URL (replace w1234567 with **YOUR ID number**)

   https://w1234567.users.ecs.westminster.ac.uk/xml_westmuni/uni_marks.xml

2. The following rendering of the XML document should be produced.

A list of modules and module leaders, and for each module a list of students on this module, as well as the marks scored in both components (mark 1 and mark 2), the final mark as an average and outcome for the module (pass or fail).



***Westmuni* Marks Report - List All Modules**

MOD101 - Database Systems			Module Leader: Francky Bandini			
Student Id	First Name	Surname	First Mark	Second Mark	Final Mark	Outcome
21051	Jim	Bim	86	80	83	PASS
21033	Kim	Li	34	42	38	FAIL
21044	Tim	Amal	62	44	53	PASS
21094	Vikash	Goode	22	30	26	FAIL
21029	Aldala	Alim	52	30	41	PASS

MOD102 - Server-side Web Development			Module Leader: Francky Bandini			
Student Id	First Name	Surname	First Mark	Second Mark	Final Mark	Outcome
21051	Jim	Bim	76	70	73	PASS
21033	Kim	Li	32	35	33.5	FAIL
21044	Tim	Amal	60	40	50	PASS
21076	Ladil	Manata	76	92	84	PASS

MOD103 - Java Programming			Module Leader: Francesco Maldini			
Student Id	First Name	Surname	First Mark	Second Mark	Final Mark	Outcome
21051	Jim	Bim	44	86	65	PASS
21033	Kim	Li	48	78	63	PASS
21044	Tim	Amal	44	65	54.5	PASS
21094	Vikash	Goode	42	30	36	FAIL
21004	Romi	Lescu	44	52	48	PASS
21017	Nagara	Kolski	38	25	31.5	FAIL

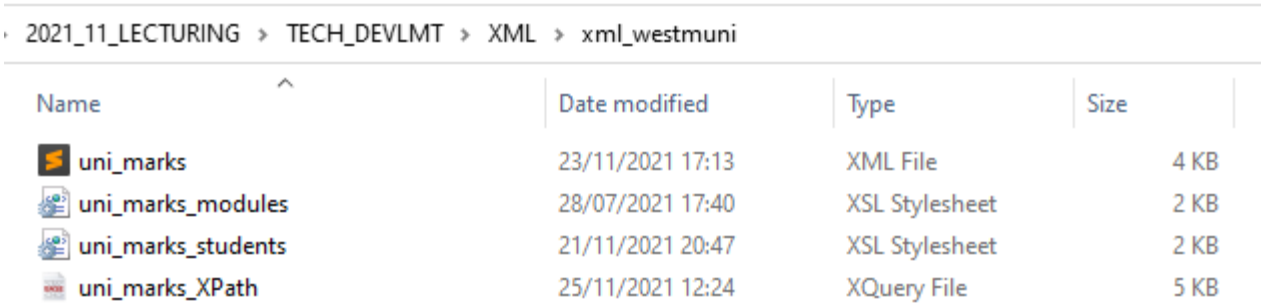
Tutorial 08 & 09 Task 03: Modify the XSLT file to render the XML document and view a list of students with their modules and marks

i. Create a new XSLT file.

1. Copy and paste **uni_marks_modules.xsl** in the same directory and rename it to **uni_marks_students.xsl**

2. Check that your **xml_westmuni directory** now contains the following files

- The XML document **uni_marks.xml**
- The XPath file **uni_marks_XPath.xqy**
- The XSLT file **uni_marks_modules.xsl**
- The XSLT file **uni_marks_students.xsl**



ii. Modify the XSLT file.

1. Edit line 2 of **uni_marks.xml** to call the new XSLT Stylesheet.

   ```
   <?xml-stylesheet type="text/xsl" href="uni_marks_students.xsl"?>
   ```

2. Open the **uni_marks_students.xsl** file in either your IDE or in XBase.

3. Modify **uni_marks_students.xsl** so that the following rendering of the XML document is produced.

   A list of all the students with the details of the modules they take, the marks they scored for both components (mark1 and mark 2), the final mark as an average, and the outcome (pass or fail).

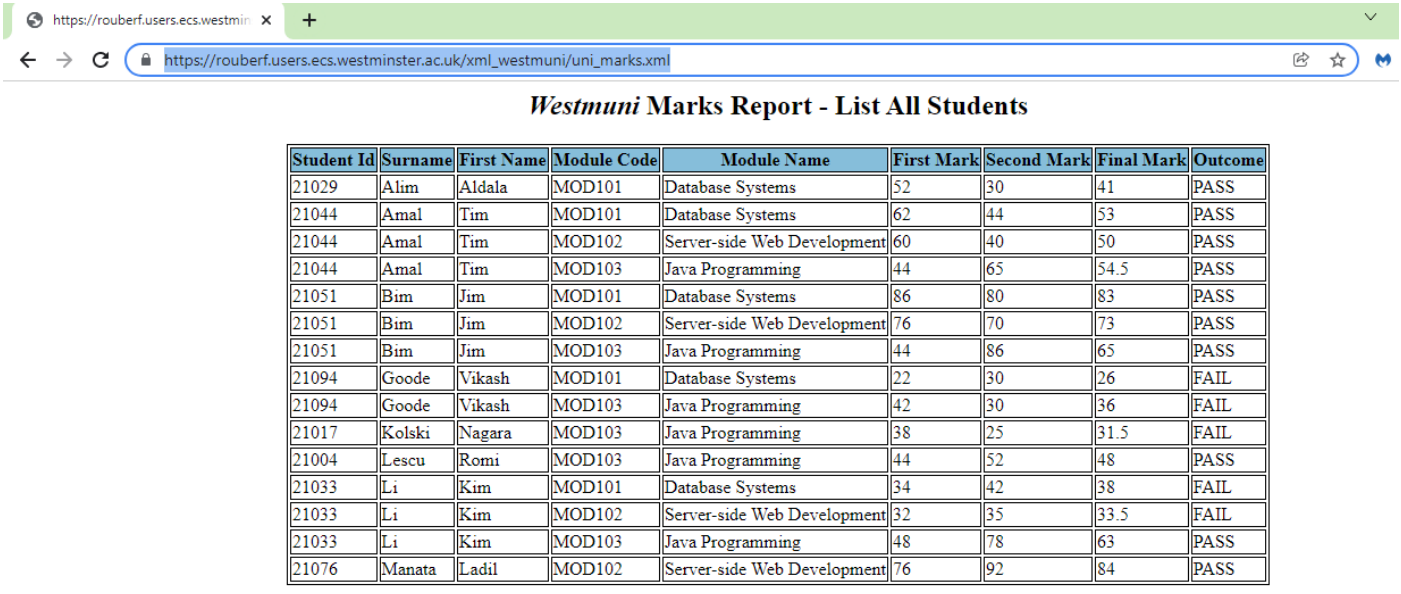
   Your list can be displayed as part of one HTML table with the following columns:

Student Id	Surname	First Name	Module Code	Module Name	1st Mark	2nd Mark	Final Mark	Outcome

   In each cell of the HTML table retrieve and display the required data valuesfrom the XML document.

4. With WinSCP (or FileZilla) re-upload all 4 files from your **local xml_westmuni directory** (left pane) to your **remote xml_westmuni directory** (right pane).

5. Refresh the browser with the same URL (with your **ID number**), you should now get the following rendering.
   https://w1234567.users.ecs.westminster.ac.uk/xml_westmuni/uni_marks.xml



***Westmuni* Marks Report - List All Students**

Student Id	Surname	First Name	Module Code	Module Name	First Mark	Second Mark	Final Mark	Outcome
21029	Alim	Aldala	MOD101	Database Systems	52	30	41	PASS
21044	Amal	Tim	MOD101	Database Systems	62	44	53	PASS
21044	Amal	Tim	MOD102	Server-side Web Development	60	40	50	PASS
21044	Amal	Tim	MOD103	Java Programming	44	65	54.5	PASS
21051	Bim	Jim	MOD101	Database Systems	86	80	83	PASS
21051	Bim	Jim	MOD102	Server-side Web Development	76	70	73	PASS
21051	Bim	Jim	MOD103	Java Programming	44	86	65	PASS
21094	Goode	Vikash	MOD101	Database Systems	22	30	26	FAIL
21094	Goode	Vikash	MOD103	Java Programming	42	30	36	FAIL
21017	Kolski	Nagara	MOD103	Java Programming	38	25	31.5	FAIL
21004	Lescu	Romi	MOD103	Java Programming	44	52	48	PASS
21033	Li	Kim	MOD101	Database Systems	34	42	38	FAIL
21033	Li	Kim	MOD102	Server-side Web Development	32	35	33.5	FAIL
21033	Li	Kim	MOD103	Java Programming	48	78	63	PASS
21076	Manata	Ladil	MOD102	Server-side Web Development	76	92	84	PASS

Model Answer for uni_marks_students.xsl

(see code overleaf)

uni_marks_students.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
  <body>
    <center><h2><i>Westmuni </i>Marks Report - List All Students</h2></center>
    <center>

     <table border="1" bordercolor="#000000" >

       <tr bgcolor="#86beda">
         <th>Student Id</th>
         <th>Surname</th>
         <th>First Name</th>
         <th>Module Code</th>
         <th>Module Name</th>
         <th>First Mark</th>
         <th>Second Mark</th>
         <th>Final Mark</th>
         <th>Outcome</th>
       </tr>

        <xsl:for-each select="marksreport/module/student">
          <xsl:sort select="sname"/>
          <xsl:sort select="fname"/>
          <tr>
             <td><xsl:value-of select="@studentId"/></td>
             <td><xsl:value-of select="sname"/></td>
             <td><xsl:value-of select="fname"/></td>
             <td><xsl:value-of select="../@moduleCode"/></td>
             <td><xsl:value-of select="../modulename"/></td>
             <td><xsl:value-of select="mark1"/></td>
             <td><xsl:value-of select="mark2"/></td>
             <td><xsl:value-of select="0.5*mark1 + 0.5*mark2"/></td>
             <td>
                <xsl:choose>
                  <xsl:when test="(0.5*mark1 + 0.5*mark2) >= 40">
                    PASS
                  </xsl:when>
                  <xsl:otherwise>
                    FAIL
                  </xsl:otherwise>
                </xsl:choose>
            </td>
          </tr>
        </xsl:for-each>

      </table>

    </center>

  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```