**Internet and Web Programming**

**Lab Fat**

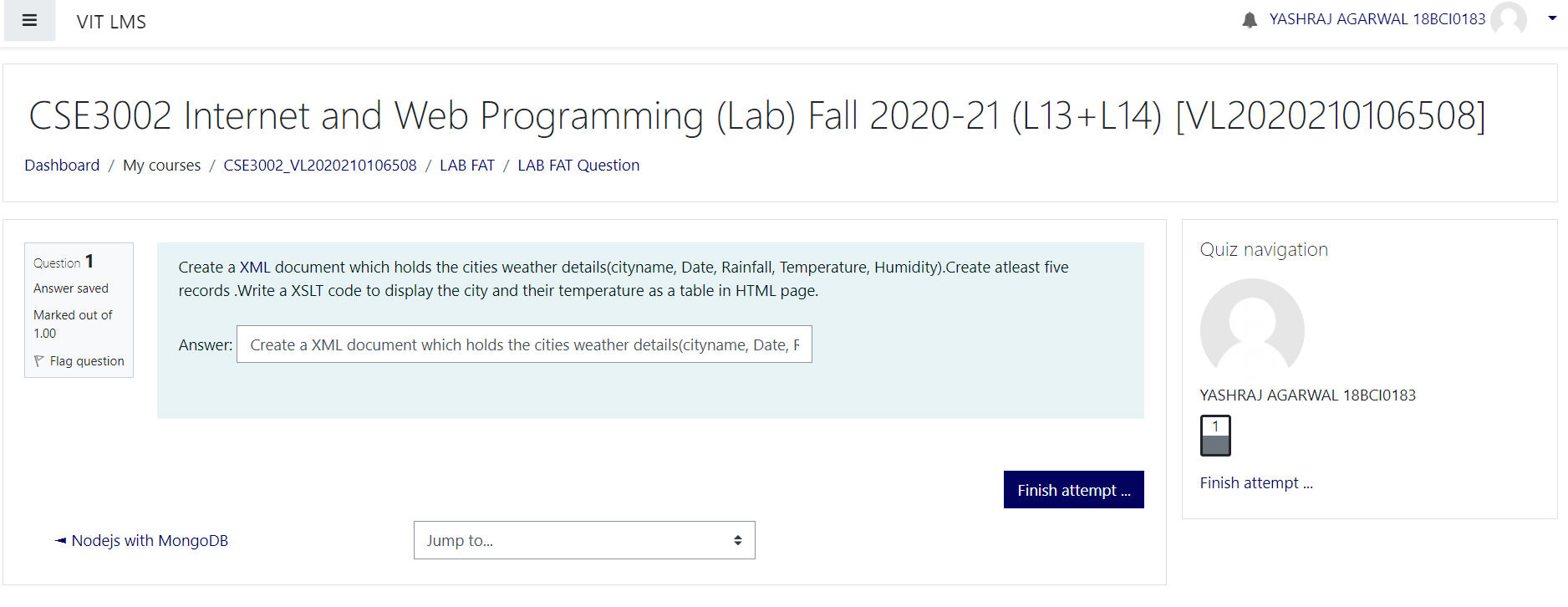
Prof. – Jayakumar K.

Name – Yashraj Agarwal

Reg. No – 18BCI0183

Slot – L13 + L14

**Question –**



**Code –**

**Records.xml**

<?xml version="1.0" ?>

<?xml-stylesheet type='text/xsl' href='studentMarks.xsl'?>

<studentMarks>

    <subjects>

        <subject>English</subject>

        <subject>Maths</subject>

    </subjects>

    <student>

        <name>Mumbai</name>

        <regno>01/11/2020</regno>

        <marks>35</marks>

    </student>

    <student>

        <name>Delhi</name>

        <regno>04/11/2020</regno>

        <marks>42</marks>

    </student>

    <student>

        <name>Bangalore</name>

        <regno>04/11/2020</regno>

        <marks>56</marks>

    </student>

    <student>

        <name>Chennai</name>

        <regno>03/11/2020</regno>

        <marks>34</marks>

    </student>

    <student>

        <name>Hyderabad</name>

        <regno>02/11/2020</regno>

        <marks>34</marks>

    </student>

</studentMarks>

**Records.xsl -**

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

<html xsl:version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    <head>

    <style>

        table, td, th {

            border: 1px solid black;

            border-collapse: collapse;

            text-align: center;

        }

    </style>

    </head>

    <body>

        <table style="width:100%;">

            <tr>

                <th rowspan="2">Name</th>

                <th rowspan="2">Date</th>

                <th rowspan="2">Temperature</th>

                <th rowspan="2">humidity</th>

                <th rowspan="2">rainfall</th>

            </tr>

            <tr>

                 <xsl:for-each select="studentMarks/subjects/subject">

                    <th><xsl:value-of select="."/></th>

                </xsl:for-each>

            </tr>

            <xsl:for-each select="studentMarks/student">

            <tr>

                <td><xsl:value-of select="name"/></td>

                <td><xsl:value-of select="regno"/></td>

                <xsl:for-each select="marks">

                    <td>

                        <xsl:value-of select="."/>

                        <xsl:text></xsl:text>

                    </td>

                </xsl:for-each>

                <td>

                    <xsl:value-of select="sum(marks)"/>

                    <xsl:text></xsl:text>

                </td>

                <td>

                    <xsl:number value="sum(marks) div count(marks)" format="1"/>

                    <xsl:text>%</xsl:text>

                </td>

                <xsl:choose>

                    <xsl:when test="(sum(marks) div count(marks)) &gt; 85">

                        <td style="background-color: green; color: white;">PASS</td>

                    </xsl:when>

                    <xsl:otherwise>

                        <td style="background-color: red; color: white;">FAIL</td>

                    </xsl:otherwise>

                </xsl:choose>

            </tr>

            </xsl:for-each>

        </table>

    </body>

</html>

**Procedure –**

|  |
| --- |
| **Creating an XML file from DTD or XML Schema** |
|  |  |
|  | Generating an XML file from a DTD or XML Schema can be useful if you want to quickly create an XML file that is based on your gramma file. |
|  | To create an XML file from a grammar file follow these steps: |
|  | 1. Invoke **New XML File** wizard using workbench menu **File>New>Other>XML>XML.** 2. On the **XML File Name** page select a project or folder to contain the XML file and type a name for it. 3. Next, depending on what type of gramma file you need choose, select the **Create XML file from DTD file** or **Create XML file from an XML Schema file** radio button. 4. The next choice you have to do is to select your gramma file. You can select it from the workspace (you can import files into the workspace if they are not there) Another possibility is to use DTD and XML files defined in **XML Catalog** . 5. On the **Select Root Element** page select from the following content options:    * Create optional attributes - both mandatory and optional attributes will be generated.    * Create optional elements - both mandatory and optional elements will be generated.    * Create first choice of required choice - the first option of a required choice will be generated in your XML file.    * Fill elements and attributes with data - any elements and attributes generated will be filled with sample data.   If you do not select any of these options, then only the minimum amount of content required for the XML file will be created.   1. The last this is to specify document type (Figure 2) for DTD or namespace (Figure 3) for XML Schema information.   For **DTD:**   * + Specify the Public ID or System ID. You do not need to specify both. If you do, the Public ID will be used before the System ID.   For **XML Schema:**   * + The Namespace information section contains information about the target namespace of the XML schema, its prefix, and the schema location |

**Output –**

