DBMS LAB 6

JEREMIAH THOMAS

106119055 | CSE A

Question 1:-

> Question 1: Getting familiar with XML Files
>
> Write a complete XML file named textbooks.xml, in which you describe at least a partial list of the textbooks you are using this semester. You should include at least two distinct textbooks. If you are using only one text this semester, expand your list to cover the current academic year. Your description of each book must include the title, author(s), publisher, year of publication, and the ISBN. For each author, specify the first and last name as distinct values. Include both a name and website for the publisher.specify the edition and cover type (paperback or hardcover) of each book you are using. Make sure your final XML document is well-formed.

Answer:-

textbook.xml

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

<jeremyLibrary>

    <book>
        <title>Operatins Systems Concepts</title>
        <author>
            <firstName>Abraham</firstName>
            <lastName>Silberschatz</lastName>
        </author>
        <publisher>
```

```
            <name>Penguin House</name>
            <website>www.penguinhouse.com</website>
        </publisher>
        <yearPublication>2007</yearPublication>
        <ISBN>3-3896-5271-3</ISBN>
        <edition>8th Edition</edition>
        <covertype>Hardbound</covertype>
    </book>

    <book>
        <title>Complex Analysis and Differential Equations</title>
        <author>
            <firstName>Kumar</firstName>
            <lastName>Sanghi</lastName>
        </author>
        <publisher>
            <name>Zondervan</name>
            <website>www.zondervan.com</website>
        </publisher>
        <yearPublication>2004</yearPublication>
        <ISBN>0-7496-5271-3</ISBN>
        <edition>8th Edition</edition>
        <covertype>Paperback</covertype>
    </book>

    <book>
        <title>Introduction to Micromanagement and Organizational
chaos</title>
        <author>
            <firstName>Ted</firstName>
            <lastName>Lasso</lastName>
        </author>
        <publisher>
```

```
            <name>Ernest Homes Publishing</name>
            <website>www.ernesthomes.com</website>
        </publisher>
        <yearPublication>2014</yearPublication>
        <ISBN>4-9496-5271-3</ISBN>
        <edition>5th Edition</edition>
        <covertype>Hardbound</covertype>
    </book>

    <book>
        <title>Machine Learning for Dummy</title>
        <author>
            <firstName>Jason</firstName>
            <lastName>Sudeikis</lastName>
        </author>
        <publisher>
            <name>O'Reilly</name>
            <website>www.oreilly.com</website>
        </publisher>
        <yearPublication>2004</yearPublication>
        <ISBN>2-3496-5271-3</ISBN>
        <edition>3rd Edition</edition>
        <covertype>Paperback</covertype>
    </book>

    <book>
        <title>Introduction To Computer Graphics</title>
        <author>
            <firstName>John</firstName>
            <lastName>Murray</lastName>
        </author>
        <publisher>
            <name>Penguin House</name>
```

```
            <website>www.penguinhouse.com</website>
        </publisher>
        <yearPublication>2006</yearPublication>
        <ISBN>6-7496-5271-3</ISBN>
        <edition>6th Edition</edition>
        <covertype>Hardcover</covertype>
    </book>

</jeremyLibrary
```

Helper Module For Questions 2 & 3:-

computeGpaModule.py

```
#Computes GPA for a student, given avg marks:-
def computeGPA(s):
    if s >= 90 and s < 100:
        return 4.5
    elif s >=85  and s < 90:
        return 4.0
    elif s >=80  and s < 85:
        return 3.7
    elif s >=77  and s < 80:
        return 3.3
    elif s >=73  and s < 77:
        return 3.0
    elif s >=70  and s < 73:
        return 2.7
    elif s >=67  and s < 70:
        return 2.5
```

```
    elif s >=63  and s < 67:
        return 2.3
    elif s >=60  and s < 63:
        return 2.0
    elif s >=50  and s < 59:
        return 1.0
    else:
        return 0.0
```

Question 2:-

> Question 2:
>
> Write a python program to read the following XML file (Given below)
> 1. Read the score of each class from score_data.xml, compute the GPA.
> 2. Add a gpa attribute for each class element.
> 3. Write the updated xml into a new file: output.xml (shown in Fiture2)
> 4. Upload your python source code file.

Code:-

- handler_q2.py

```
import xml.etree.ElementTree as ET
import computeGpaModule

tree = ET.parse('score_data.xml')
root = tree.getroot()

for curStudent in root.findall('student'):

    print(curStudent.get('student_id'))
```

```
    score = int(curStudent.find('xml_class').text)
    print("xml", score)
    curStudent.find('xml_class').set('gpa',
str(computeGpaModule.computeGPA(score)))
    print("DS", score)
    score = int(curStudent.find('data_structure').text)
    curStudent.find('data_structure').set('gpa',
str(computeGpaModule.computeGPA(score)))
    print("Algorithm", score)
    score = int(curStudent.find('algorithm').text)
    curStudent.find('algorithm').set('gpa',
str(computeGpaModule.computeGPA(score)))
    print("Network", score)
    score = int(curStudent.find('network').text)
    curStudent.find('network').set('gpa',
str(computeGpaModule.computeGPA(score)))

#Write into output file
tree.write('output.xml')
```

Output:-

- output.xml

```
<score_data>
    -
    <student student_id="A001">
        <xml_class gpa="2.0">60</xml_class>
        <data_structure gpa="2.7">70</data_structure>
        <algorithm gpa="4.0">85</algorithm>
        <network gpa="4.5">90</network>
    </student>
    -
    <student student_id="A002">
```

```
        <xml_class gpa="2.3">66</xml_class>
        <data_structure gpa="3.3">78</data_structure>
        <algorithm gpa="2.0">62</algorithm>
        <network gpa="4.0">88</network>
    </student>
    -
    <student student_id="A003">
        <xml_class gpa="4.0">89</xml_class>
        <data_structure gpa="3.3">77</data_structure>
        <algorithm gpa="3.7">80</algorithm>
        <network gpa="1.0">50</network>
    </student>

</score_data>
```

Question 3:-

Question 3:

Write a python program to read the XML file output.xml (given below).
1. Add 5 to each class's score, compute the GPA.
2. Compute the average GPA.
3. Add average element for average GPA of each student
4. Write the updated xml into a new file: new_output.xml

Code:-

- handler_q3.py

```
import xml.etree.ElementTree as ET
import computeGpaModule

tree = ET.parse('output.xml')
```

```
root = tree.getroot()
avgScore=0

for curStudent in root.findall('student'):
    item = curStudent.find('xml_class')
    score = int(item.text)
    score+=5
    avgScore=score
    item.text = str(score)
    item.set('gpa', str(computeGpaModule.computeGPA(score)))

    item = curStudent.find('data_structure')
    score = int(item.text)
    score+=5
    avgScore+=score
    item.text = str(score)
    item.set('gpa', str(computeGpaModule.computeGPA(score)))

    item = curStudent.find('algorithm')
    score = int(item.text)
    score+=5
    avgScore+=score
    item.text = str(score)
    item.set('gpa', str(computeGpaModule.computeGPA(score)))

    item = curStudent.find('network')
    score = int(item.text)
    score+=5
    avgScore+=score
    item.text = str(score)
    item.set('gpa', str(computeGpaModule.computeGPA(score)))

    student_id = curStudent.get('student_id')
```

```
    avgScore = avgScore/4
    print("For Student : " +  str(student_id) + " , Avg Sum is: " +
str(avgScore))
    print("GPA is: " + str(computeGpaModule.computeGPA(avgScore)))

    avgGPA = ET.SubElement(curStudent, 'avgGPA')
    avgGPA.text = str(computeGpaModule.computeGPA(avgScore))

tree.write('new_output.xml')
```

Output:-

- new_output.xml

```
<score_data>
    -
    <student student_id="A001">
        <xml_class gpa="2.3">65</xml_class>
        <data_structure gpa="3.0">75</data_structure>
        <algorithm gpa="4.5">90</algorithm>
        <network gpa="4.5">95</network>
        <avgGPA>3.7
 </avgGPA></student>
    -
    <student student_id="A002">
        <xml_class gpa="2.7">71</xml_class>
        <data_structure gpa="3.7">83</data_structure>
        <algorithm gpa="2.5">67</algorithm>
        <network gpa="4.5">93</network>
        <avgGPA>3.3</avgGPA>
    </student>
    -
    <student student_id="A003">
        <xml_class gpa="4.5">94</xml_class>
```

```
        <data_structure gpa="3.7">82</data_structure>
        <algorithm gpa="4.0">85</algorithm>
        <network gpa="1.0">55</network>
        <avgGPA>3.3</avgGPA>
    </student>

</score_data>
```

THANK YOU!