Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра Штучного інтелекту

Дисципліна: «Інтелектуальні технології в Internet та Semantic Web»

Лабораторна робота № 2

на тему: «ПРОГРАМНА ОБРОБКА XML-ДОКУМЕНТІВ»

|  |  |  |
| --- | --- | --- |
| Виконали:  ст. гр. ІТКН-17-7  Савенков Д.В. |  | Перевірив:  Бібічков І. Є. |
|  |  |  |

Харків 2020

2 ПРОГРАМНА ОБРОБКА XML-ДОКУМЕНТІВ

**2.1 Мета роботи**

Використання DOM (Document Object Model) для програмного створення і обробки XML-документів.

**Варіант -** 'https://djinni.co/set\_lang?code=en&next=/developers/' (резюме ІТ-спеціалістів)

**Посилання на гіт-репозиторій:** https://github.com/denys-savenkov/Savenkov-Semantic-Web

**2.2 Завдання**

1. На будь-якій мові програмування (рекомендується мова Java) реалізувати програму, яка модифікує XML-документ із лабораторної роботи №1. Рекомендується використовувати класи для об’єктів із XML-документу.

2. Для XML-документу необхідно програмно реалізувати:

- модифікування текстових значень елементів (додавання/видалення) (2-3 значення)

- модифікування атрибутів складних елементів (додавання/видалення) (2-3 значення)

- додавання складних ієрархій типів елементів (1-2 типу);

3. Сгенерувати програмним шляхом HTML-документ за XML-схемою та XSL-таблицею стилів.

**2.2.1 Аргументація вибору мови програмування та засобів розробки**

Для виконання поставленого практичного завдання була обрана мова програмування Python за її легкість, зручність, наявність великої кількості бібліотек та добре складеної документації. Серед засобів розробки була обрана IDE PyCharm за її функціонал, зокрема, інтеграції механізмів та автоматизації роботи із XML та XSL файлами.

**2.3 Схематичне зображення DOM оригінального та модифікованого XML-документу**

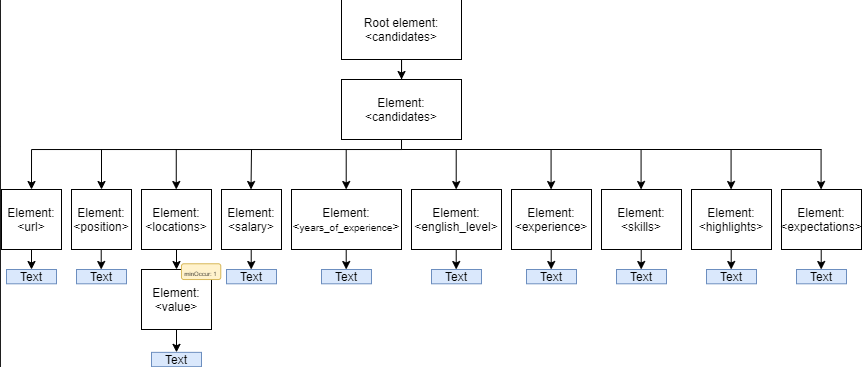


Рис. 2.1 – DOM оригінального XML

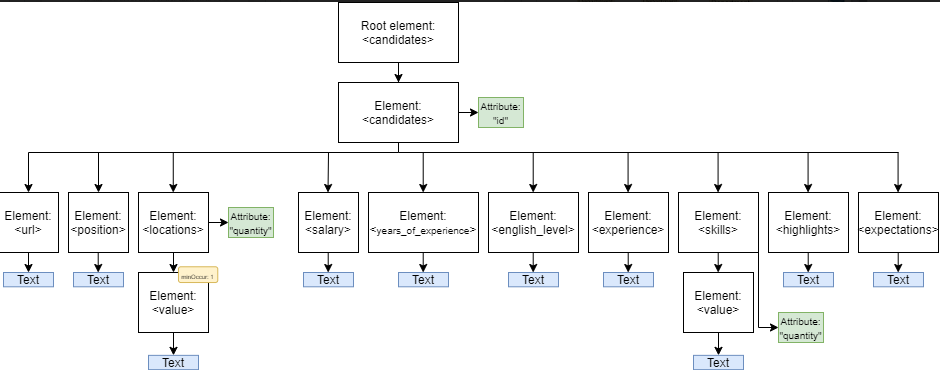


Рис 2.2 – DOM модифікованогоXML

Також зокрема зазначених на рис. 2.1 та рис. 2.2 змін, були оновлені наступні текстові значення:

* З елементів <salary> були убрані знаки доларів, так як усі значення були наведені у однакової валюті (наприклад, було «$800», стало – «800»)
* Значення елементів <years\_of\_experience> були переведені з дробових чисел до натуральних (з округленням у меншу сторону, наприклад, було «4.5», стало – «4»)

**2.4 XSL-таблиця стилів**

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

<xsl:stylesheet version="1.0"

xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<html>

<body>

<h2>Djinni.co Candidates List</h2>

<table border="1">

<tr bgcolor="#9acd32">

<th>url</th>

<th>position</th>

<th>locations</th>

<th>salary</th>

<th>years\_of\_experience</th>

<th>english\_level</th>

<th>experience</th>

<th>skills</th>

<th>highlights</th>

<th>expectations</th>

</tr>

<xsl:for-each select="candidates/candidate">

<tr>

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

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

<td><xsl:apply-templates select="locations" /></td>

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

<td><xsl:value-of select="years\_of\_experience"/></td>

<td><xsl:value-of select="english\_level"/></td>

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

<td><xsl:apply-templates select="skills" /></td>

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

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

</tr>

</xsl:for-each>

</table>

</body>

</html>

</xsl:template>

<xsl:template match="locations">

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

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

</xsl:for-each>

</xsl:template>

<xsl:template match="skills">

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

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

</xsl:for-each>

</xsl:template>

</xsl:stylesheet>

**2.5 Програмний код для створення HTML-документу**

import lxml.etree as ET

from lxml.html.clean import unicode

dom = ET.parse('new\_djinni.xml')

xslt = ET.parse('candidates\_list.xsl')

transform = ET.XSLT(xslt)

newdom = transform(dom)

print(ET.tostring(newdom, pretty\_print=True))

print(newdom)

with open("candidates.html", 'w', encoding='utf-8') as outfile:

outfile.write(str(newdom))

**2.6 Скріншот вікна браузера із відображенням створеної веб-сторінки.**

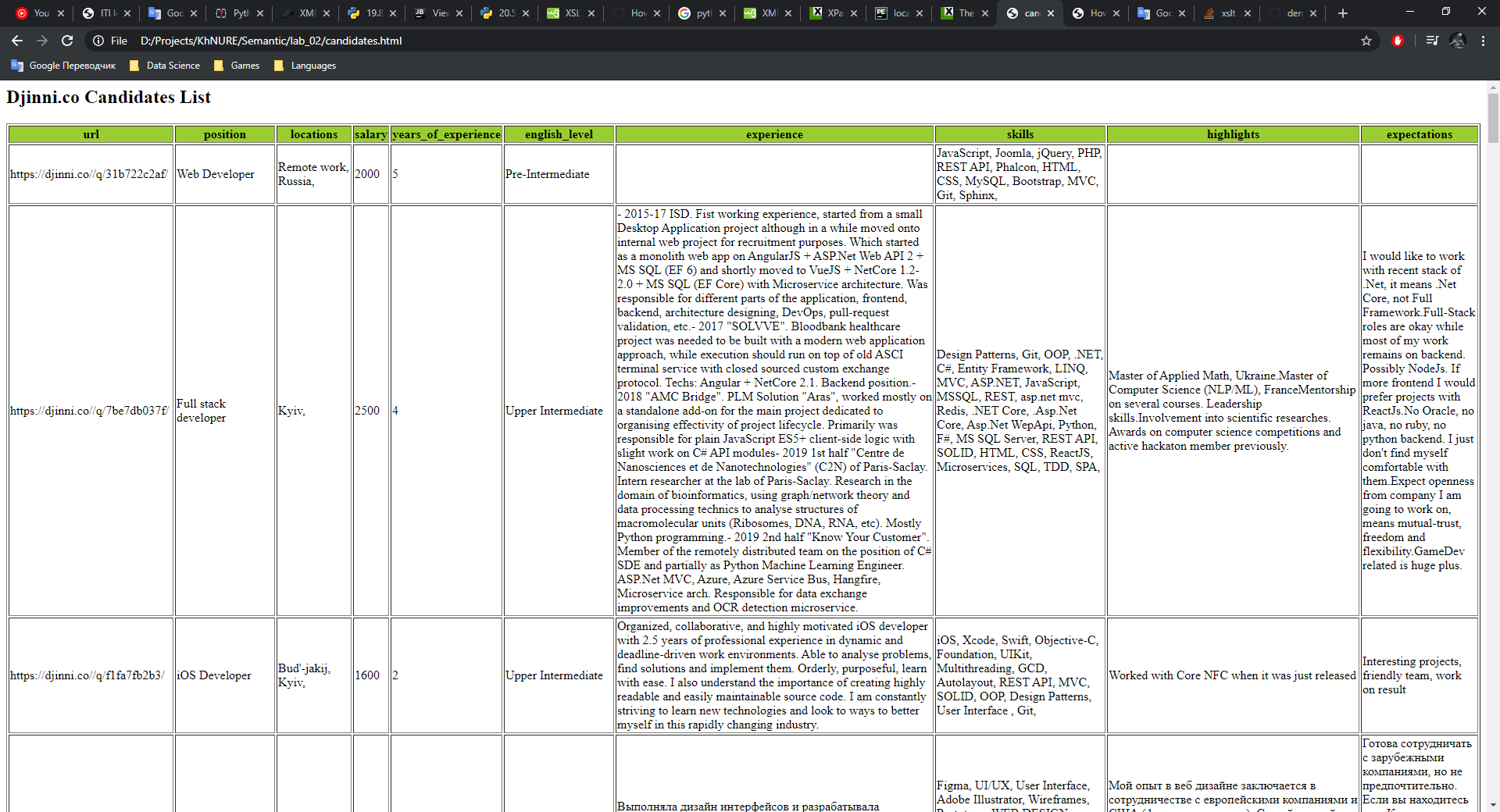


Рис. 2.3 – Скріншот вікна

**2.7 Висновки**

У результаті виконання даної практичної роботи були засвоєні основні принципи роботи із DOM XML-документів, що дозволяє програмний шляхом (наприклад, використовуючи мову програмування Python) працювати з вмістом файлу, а також виконувати різні операції: міняти або видаляти існуючи елементи та їх атрибути та створювати нові при необхідності.

Окрім DOM, був розглянутий механізм роботи XSLT файлу, що є «файлом-інтерфейсом», який дозволяє швидко програмний шляхом перевести вміст xml до файлу html та стилями css, що дозволяє інтегрувати та відображати дані онлайн у браузері.

**2.8 Контрольні запитання**

1) Що таке об’єктна модель документа?

DOM – логічна модель XML документу, що відображає його структуру та є інтерфейсом, що дозволяє програмам та скриптам взаємодіяти із документами сімейства XML (XML, HTML, XHTML), переглядати й маніпулювати із даними та міняти логічну структуру. Даний інтерфейс є незалежним від платформи, ОС або мови програмування.

2) Для чого використовується об’єктна модель документа?

Для маніпуляції з внутрішнім вмістом xml-файлів та для зміни (додавання, видалення елементів або атрибутів) структури

3) Яким чином будується об’єктна модель XML-документа?

DOM модель будується у деревовидній структурі, де кожний вузол є елементом xml документа, а дочірні вузли – вкладені елементи. Також вузли зберігають текстову інформацію елементу та значення його атрибутів

4) Опишіть послідовність дій для створення HTML-документу на основі даних, що зберігають в XML.

* Створити та заповнити XML документ (бажано, провалідувати використовуючи DTD- та/або XSD-схему)
* Створити XSL-таблицю стилів, де прописати структуру майбутнього html-документу та призначити, використовуючи DOM, до нього елементи XML. При необхідності, прописати усі необхідні CSS стилі
* Розробити програмний код або використати існуючий для перетворення XML у HTML через «призму» XSL-схеми