ASSIGNMENT NO# 03

NAME: DANYAL SHAH

REG. NO: 200901046

SECTION: B

SUBJECT: Compiler Construction

DATE: 14-01-2023

SUBMITTED TO: Ms. REEDA SAEED

TASK: In this Assignment we are have to parse an xml file in python and extract following data from it.

1.Book_Id

2.Author_Name

3.Title

4.Genre

5.Price

6.Publish_Date

7.Description

After the first part we are to Create an Excel file to record the data extracted from the xml file and organize it according to the specified format.

Objective:

In this Assignment We have to parse an XML file using any programming language.I have used Python language for this task.The xml data provided is named as “compiler.xml” and our objective is to extract the data the data from it and read accordingly.

The library used for this purpose is the xml.etree.ElementTree which is a built-in Python library for parsing XML and HTML files.It provides a simple way to navigate and modify the elements and attributes of an XML document.

Source Code:

```
#Danyal Shah
#200901046
#Parsing XML file using python and extracting the data to an excel file

import xml.etree.ElementTree as ElementTree
import pandas as pd

tree = ElementTree.parse('compiler.xml')
root = tree.getroot()

# Create an empty list to store the data
data = []

# Iterate over the 'book' elements
for book in root.findall('book'):
    id = book.get('id')
    author = book.find('author').text
    title = book.find('title').text
    genre = book.find('genre').text
    try:
        price = float(book.find('price').text)
    except ValueError:
        price = None
    publish_date = book.find('publish_date').text
    description = book.find('description').text

    print("Book ID: ", id)
    print("Author: ", author)
    print("Title: ", title)
    print("Genre: ", genre)
    print("Price: ", price)
    print("Publish Date: ", publish_date)
    print("Description: ", description)
    print("\n")
    print("-"*50)

    # Append the data to the list
    data.append([id, author, title, genre, price, publish_date, description])

# Create a dataframe from the list
    df = pd.DataFrame(data, columns=['Book_ID', 'Author', 'Title', 'Genre', 'Price',
'Publish_Date', 'Description'])
```

```
# Write the dataframe to an Excel file
df.to_excel('200901046_Assign_03.xlsx', index=False)
```

Explanation:

1. Importing **xml.etree.ElementTree as ElementTree** : The first line of the code imports the ElementTree module from the xml.etree package, which allows the code to parse and manipulate XML files.
2. After that we import the **pandas package module** which is used for data manipulation and analysis.
3. **Parsing XML file:** The code uses the parse() function from ElementTree to parse the 'compiler.xml' file, and assigns the resulting ElementTree object to the variable 'tree'.

4. **Accessing the root of the tree**: The getroot() method from the ElementTree object is used to access the root element of the XML file and assigns the resulting element to the variable 'root'.
5. **List**: We declare an empty list to store the data in it
6. **Iterating over the book element**: The findall() method from the 'root' element is used to find all of the 'book' elements in the XML file, and assigns the resulting list of elements to the variable 'book'. Then it iterates over the 'book' elements.
7. **Extracting the information**: Inside the for loop, the code uses the get() method to extract the 'id' attribute of the 'book' element and assigns it to the variable 'id', then it uses the find() method to extract the 'author', 'title', 'genre', 'price', 'publish_date', and 'description' elements of the 'book' element and assigns the resulting text values to the corresponding variables
8. **Exception Handling**:The try and error method declares the datatype of the Price attribute to be a float in order to handle exceptions.The try block attempts is used so that only numbers can be extracted in the price section in excel file and any non numerical values will be handled properly to convert the text of the price element to a float
9. **Printing the extracted information:** The code then prints the extracted information, Author, Title, Genre, Price

10. The pandas library is used to create a dataframe, add data to it, and then write the dataframe to an Excel file.

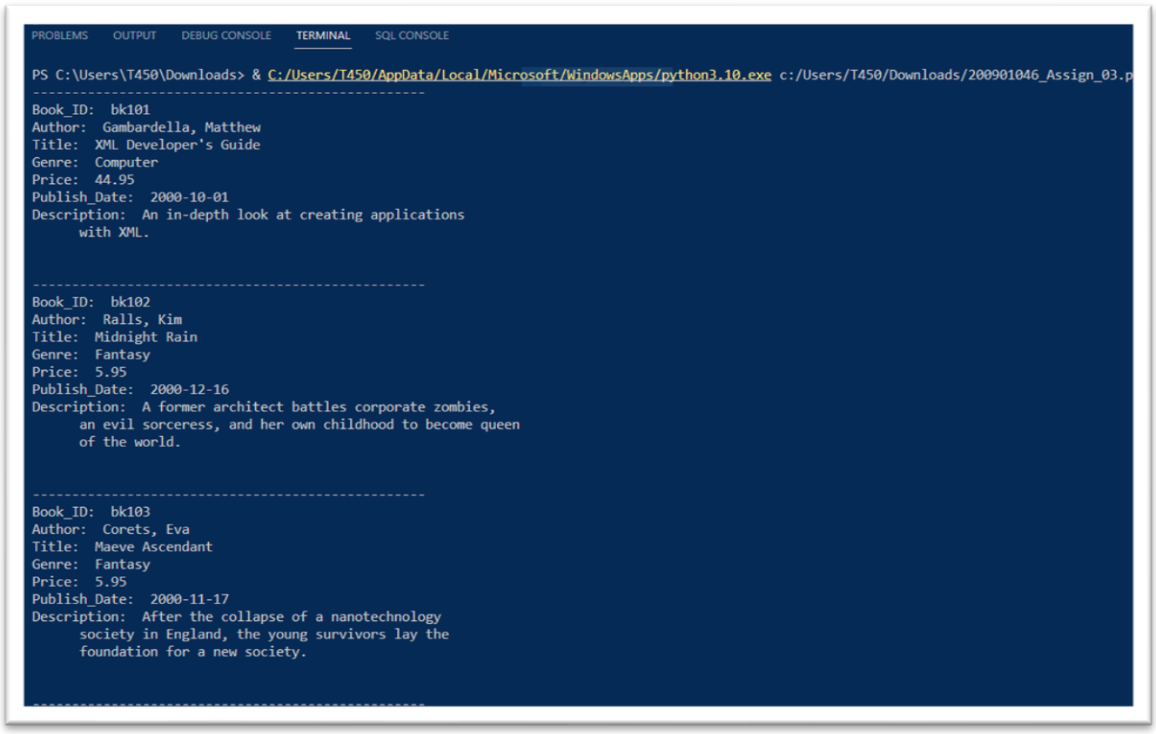
11. 11**.append()** method adds data to the list, containing an id, author, title, genre, price, publish_date, and description.

12. Then a dataframe is created, df, from the data list, and the column names are setted as 'Book_ID', 'Author', 'Title', 'Genre', 'Price', 'Publish_Date', 'Description'.

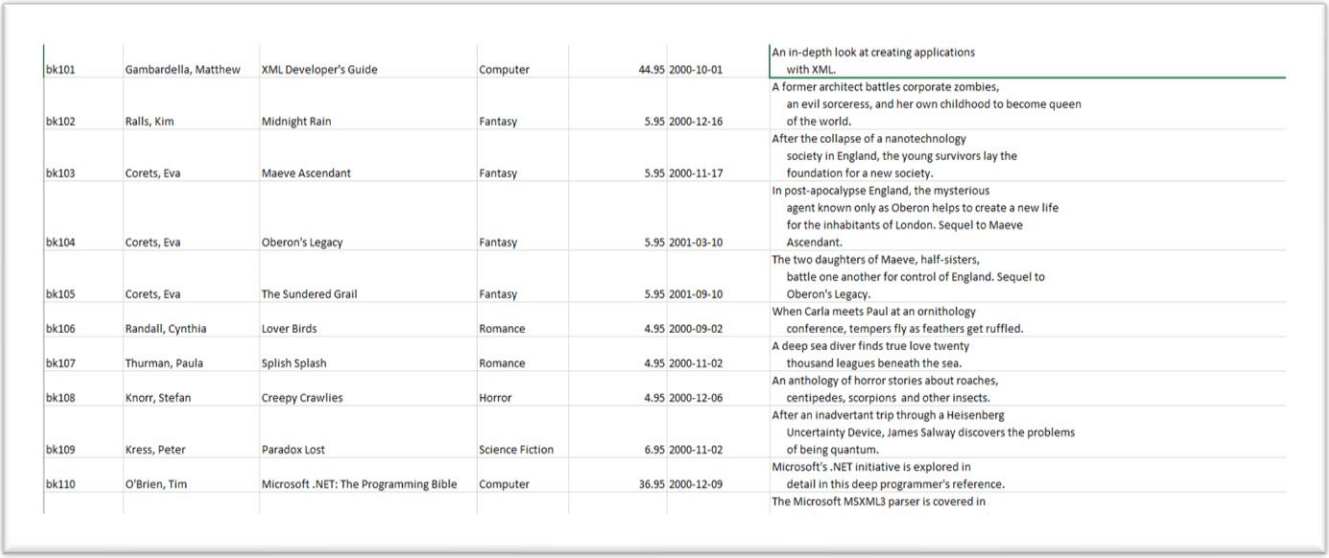
13. The last line is using the to_excel() method to write the dataframe to an Excel file named '200901046_Assign_03.xlsx' and not to write the index of dataframe.

Output Screenshot:

The output displays the extracted data from the xml data showing all the attributes of the Book.



Output of Excel File:



bk101	Gambardella, Matthew	XML Developer's Guide	Computer	44.95	2000-10-01	An in-depth look at creating applications with XML.
bk102	Ralls, Kim	Midnight Rain	Fantasy	5.95	2000-12-16	A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.
bk103	Corets, Eva	Maeve Ascendant	Fantasy	5.95	2000-11-17	After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.
bk104	Corets, Eva	Oberon's Legacy	Fantasy	5.95	2001-03-10	In post-apocalypse England, the mysterious agent known only as Oberon helps to create a new life for the inhabitants of London. Sequel to Maeve Ascendant.
bk105	Corets, Eva	The Sundered Grail	Fantasy	5.95	2001-09-10	The two daughters of Maeve, half-sisters, battle one another for control of England. Sequel to Oberon's Legacy.
bk106	Randall, Cynthia	Lover Birds	Romance	4.95	2000-09-02	When Carla meets Paul at an ornithology conference, tempers fly as feathers get ruffled.
bk107	Thurman, Paula	Splish Splash	Romance	4.95	2000-11-02	A deep sea diver finds true love twenty thousand leagues beneath the sea.
bk108	Knorr, Stefan	Creepy Crawlies	Horror	4.95	2000-12-06	An anthology of horror stories about roaches, centipedes, scorpions and other insects.
bk109	Kress, Peter	Paradox Lost	Science Fiction	6.95	2000-11-02	After an inadvertant trip through a Heisenberg Uncertainty Device, James Salway discovers the problems of being quantum.
bk110	O'Brien, Tim	Microsoft .NET: The Programming Bible	Computer	36.95	2000-12-09	Microsoft's .NET initiative is explored in detail in this deep programmer's reference.
						The Microsoft MSXML3 parser is covered in

Github Link:

danyalthewebdev/Parsing-an-XML-File-using-Python: Parsing an XML file using python and extracting the data to an excel file (github.com)