Trinity College Dublin

Coláiste na Tríonóide, Baile Átha Cliath

The University of Dublin

Knowledge Engineering

FINAL ASSIGNMENT

Declaration: "I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar. I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at http://tcdie.libguides.com/plagiarism/ready-steady-write."

Submitted By

Harsh Dhingra

19323904

Q1

(a) Develop a DTD to describe a new XML vocabulary of your choosing, e.g. a music collection with artists and albums (but please choose a different example). The DTD should specify both elements and attributes. Element nesting and cardinality of some elements should also be included. Use as many features of DTD as you deem appropriate.

[5 Marks]

The below DTD represents an XML vocabulary for a Restaurant. Basically, a DTD is a set of markup definitions. DTD can be defined as a building blocks of XML. The Restaurant contains Food, beverage. Food contains starters, main_courses and Protein_Shakes. The starter has a cardinality of One-or-more. The starter further hierarchy of name, price , ingredients, calories, macros. Salad, Gym_diet, shake, diet_drinks also has cardinality of one or more i.e +.

```
<!ELEMENT Restaurant (Food, beverage)>
<!ELEMENT Food (starters, main_courses, Protein_Shakes)>
<!ELEMENT starters (starter+)>
<!ELEMENT starter (name, price, ingredients, calories, macros)>
<!ELEMENT main_courses (salads, Gym_diets)>
<!ELEMENT salads (salad+)>
<!ELEMENT salad (name, price, ingredients, calories, macros)>
<!ELEMENT Gym_diets (Gym_diet+)>
<!ELEMENT Gym_diet (name, price, weight?, ingredients, calories, macros)>
<!ELEMENT Protein_Shakes (shake+)>
<!ELEMENT shake (name, price, ingredients, calories, macros)>
<!ELEMENT beverage (diet_drinks)>
<!ELEMENT diet_drinks (diet_drink+)>
<!ELEMENT diet_drink (name, price, calories, volume*)>
<!ELEMENT macros (protein, fat, carbs)>
<!ELEMENT price (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT ingredients (#PCDATA)>
<!ELEMENT calories (#PCDATA)>
<!ELEMENT protein (#PCDATA)>
<!ELEMENT fat (#PCDATA)>
<!ELEMENT carbs (#PCDATA)>
<!ELEMENT weight (#PCDATA)>
<!ELEMENT volume (#PCDATA)>
<!ATTLIST starter id CDATA #REQUIRED>
<!ATTLIST salad id CDATA #REQUIRED>
<!ATTLIST Gym_diet id CDATA #REQUIRED>
<!ATTLIST shake id CDATA #REQUIRED>
<!ATTLIST diet_drink id CDATA #REQUIRED>
<!ATTLIST price currency CDATA #REQUIRED>
<!ATTLIST calories unit CDATA #FIXED "kcal">
<!ATTLIST protein unit CDATA #FIXED "g">
<!ATTLIST fat unit CDATA #FIXED "g">
<!ATTLIST carbs unit CDATA #FIXED "g">
<!ATTLIST weight unit CDATA #FIXED "gram">
<!ATTLIST volume unit CDATA #FIXED "ml">
```

(b) Create a XSD version of the DTD from part (a). Feel free to augment the vocabulary by leveraging some of the advanced features of XSD. Illustrate how leveraging these features has enabled more rigorous validation of XML documents.

[10 Marks]

XSD is an XML schema definition, and it defines formally the elements and attributes in XML. Its created with the help of DTD, and it helps in the validation of XML.

Now XSD gives a clear hierarchy, and various cardinalities are being defined as Minimum occurrence and Maximum occurrences. As the name suggests, minimum tells us the minimum no. of values for an element, and if Maximum occurrence is set as Unbounded, it means we can have N number of values. XSD helps us define default values to an attribute.

I have added minimum occurrence and maximum occurrence to every element, and all attributes are set as required so nothing can be deleted/ missed. Once the XSD and XML are connected, if there is an error, it won't validate XML. Therefore these help in the rigorous validation of XML documents. For example, if element id is set as required and not present in XML, it won't validate.

5/27/2021 Restau.xsd

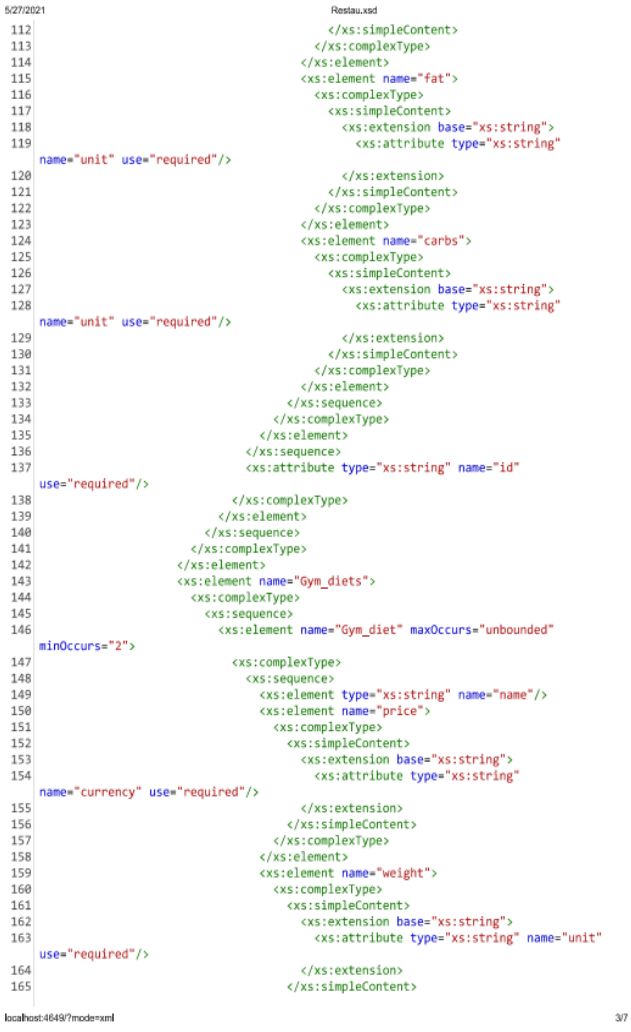
```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Restaurant">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Food">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="starters">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="starter" maxOccurs="unbounded" minOccurs="3">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element type="xs:string" name="name"/>
                          <xs:element name="price">
                            <xs:complexType>
                              <xs:simpleContent>
                                <xs:extension base="xs:string">
                                  <xs:attribute type="xs:string" name="currency"
use="required"/>
                                </xs:extension>
                              </xs:simpleContent>
                            </xs:complexType>
                          </xs:element>
                          <xs:element type="xs:string" name="ingredients"/>
                          <xs:element name="calories">
                            <xs:complexType>
                              <xs:simpleContent>
                                <xs:extension base="xs:string">
                                  <xs:attribute type="xs:string" name="unit"
use="required"/>
                                </xs:extension>
                              </xs:simpleContent>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="macros">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="protein">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:string">
                                        <xs:attribute type="xs:string" name="unit"
use="required"/>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="fat">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:string">
                                        <xs:attribute type="xs:string" name="unit"
use="required"/>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="carbs">
                                  <xs:complexType>
```

localhost:4649/?mode=xml 1/7

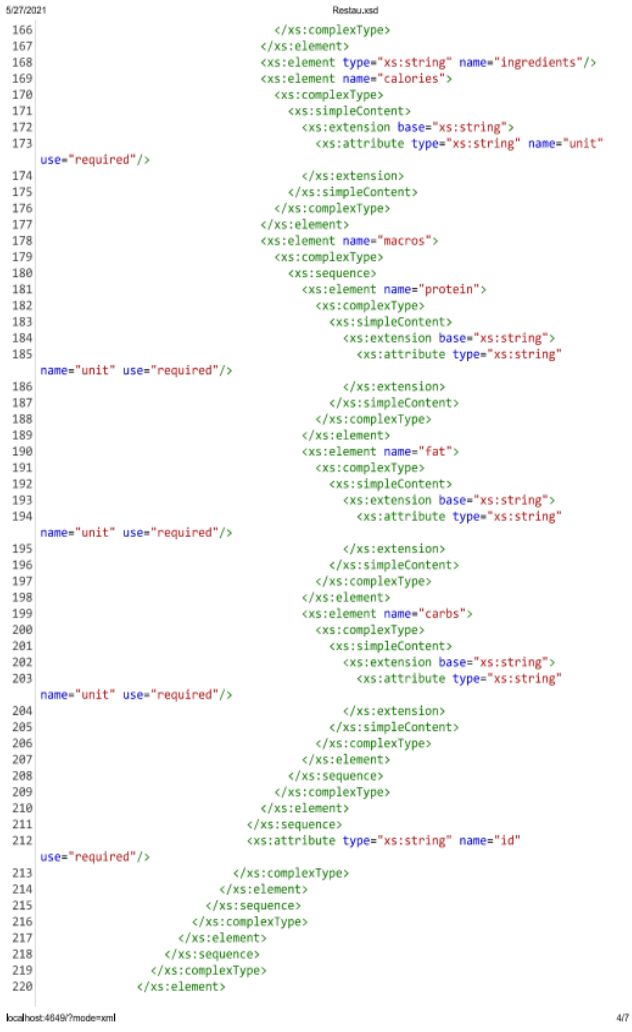
5/27/2021 Restau.xsd

```
                                    <xs:simpleContent>
                                      <xs:extension base="xs:string">
                                        <xs:attribute type="xs:string" name="unit"
use="required"/>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                          <xs:attribute type="xs:string" name="id" use="required"/>
                        </xs:complexType>
                      </xs:element>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                <xs:element name="main_courses">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="salads">
                        <xs:complexType>
                          <xs:sequence>
                            <xs:element name="salad" maxOccurs="unbounded"
minOccurs="3">
                              <xs:complexType>
                                <xs:sequence>
                                  <xs:element type="xs:string" name="name"/>
                                  <xs:element name="price">
                                    <xs:complexType>
                                      <xs:simpleContent>
                                        <xs:extension base="xs:string">
                                          <xs:attribute type="xs:string"
name="currency" use="required"/>
                                        </xs:extension>
                                      </xs:simpleContent>
                                    </xs:complexType>
                                  </xs:element>
                                  <xs:element type="xs:string" name="ingredients"/>
                                  <xs:element name="calories">
                                    <xs:complexType>
                                      <xs:simpleContent>
                                        <xs:extension base="xs:string">
                                          <xs:attribute type="xs:string" name="unit"
use="required"/>
                                        </xs:extension>
                                      </xs:simpleContent>
                                    </xs:complexType>
                                  </xs:element>
                                  <xs:element name="macros">
                                    <xs:complexType>
                                      <xs:sequence>
                                        <xs:element name="protein">
                                          <xs:complexType>
                                            <xs:simpleContent>
                                              <xs:extension base="xs:string">
                                                <xs:attribute type="xs:string"
name="unit" use="required"/>
                                              </xs:extension>
```

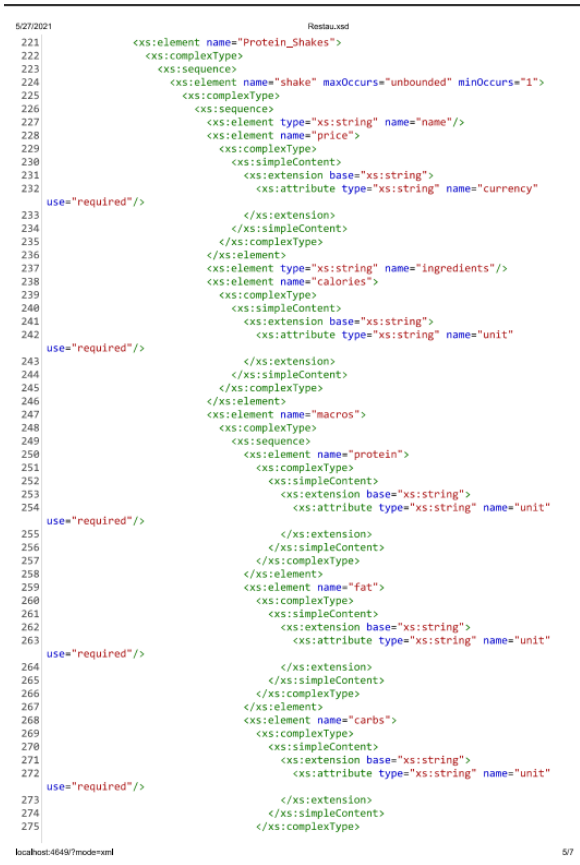
localhost:4649/?mode=xml 2/7



```
5/27/2021                                   Restau.xsd
                                          </xs:simpleContent>
                                        </xs:complexType>
                                      </xs:element>
                                      <xs:element name="fat">
                                        <xs:complexType>
                                          <xs:simpleContent>
                                            <xs:extension base="xs:string">
                                              <xs:attribute type="xs:string"
name="unit" use="required"/>
                                            </xs:extension>
                                          </xs:simpleContent>
                                        </xs:complexType>
                                      </xs:element>
                                      <xs:element name="carbs">
                                        <xs:complexType>
                                          <xs:simpleContent>
                                            <xs:extension base="xs:string">
                                              <xs:attribute type="xs:string"
name="unit" use="required"/>
                                            </xs:extension>
                                          </xs:simpleContent>
                                        </xs:complexType>
                                      </xs:element>
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                              <xs:attribute type="xs:string" name="id"
use="required"/>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="Gym_diets">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Gym_diet" maxOccurs="unbounded"
minOccurs="2">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element type="xs:string" name="name"/>
                                <xs:element name="price">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:string">
                                        <xs:attribute type="xs:string"
name="currency" use="required"/>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="weight">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:string">
                                        <xs:attribute type="xs:string" name="unit"
use="required"/>
                                      </xs:extension>
                                    </xs:simpleContent>
localhost:4649/?mode=xml                                   3/7
```



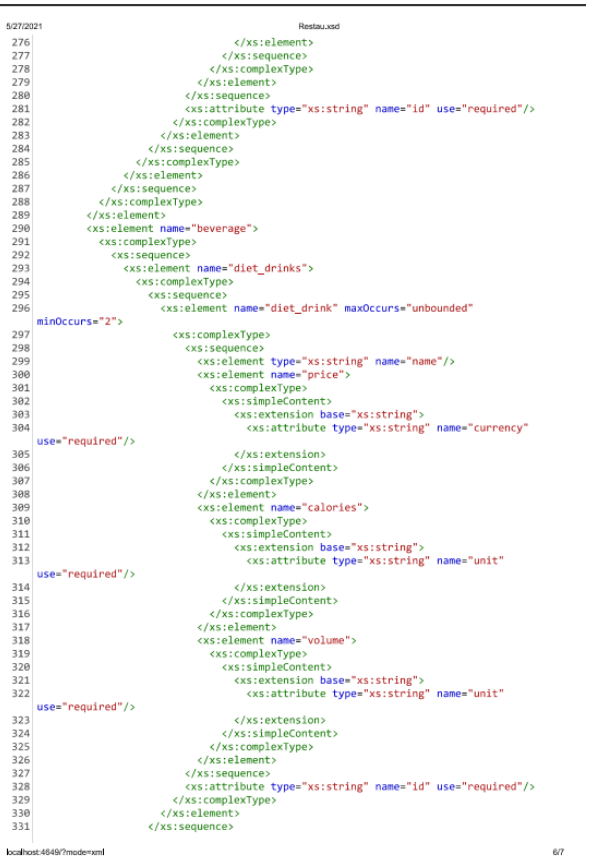
```
5/27/2021                                   Restau.xsd
                                  </xs:complexType>
                                </xs:element>
                                <xs:element type="xs:string" name="ingredients"/>
                                <xs:element name="calories">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:string">
                                        <xs:attribute type="xs:string" name="unit"
use="required"/>
                                      </xs:extension>
                                    </xs:simpleContent>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="macros">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="protein">
                                        <xs:complexType>
                                          <xs:simpleContent>
                                            <xs:extension base="xs:string">
                                              <xs:attribute type="xs:string"
name="unit" use="required"/>
                                            </xs:extension>
                                          </xs:simpleContent>
                                        </xs:complexType>
                                      </xs:element>
                                      <xs:element name="fat">
                                        <xs:complexType>
                                          <xs:simpleContent>
                                            <xs:extension base="xs:string">
                                              <xs:attribute type="xs:string"
name="unit" use="required"/>
                                            </xs:extension>
                                          </xs:simpleContent>
                                        </xs:complexType>
                                      </xs:element>
                                      <xs:element name="carbs">
                                        <xs:complexType>
                                          <xs:simpleContent>
                                            <xs:extension base="xs:string">
                                              <xs:attribute type="xs:string"
name="unit" use="required"/>
                                            </xs:extension>
                                          </xs:simpleContent>
                                        </xs:complexType>
                                      </xs:element>
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                              <xs:attribute type="xs:string" name="id"
use="required"/>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
localhost:4649/?mode=xml                                   4/7
```



5/27/2021 Restau.xsd

```
221              <xs:element name="Protein_Shakes">
222                <xs:complexType>
223                  <xs:sequence>
224                    <xs:element name="shake" maxOccurs="unbounded" minOccurs="1">
225                      <xs:complexType>
226                        <xs:sequence>
227                          <xs:element type="xs:string" name="name"/>
228                          <xs:element name="price">
229                            <xs:complexType>
230                              <xs:simpleContent>
231                                <xs:extension base="xs:string">
232                                  <xs:attribute type="xs:string" name="currency"
    use="required"/>
233                                </xs:extension>
234                              </xs:simpleContent>
235                            </xs:complexType>
236                          </xs:element>
237                          <xs:element type="xs:string" name="ingredients"/>
238                          <xs:element name="calories">
239                            <xs:complexType>
240                              <xs:simpleContent>
241                                <xs:extension base="xs:string">
242                                  <xs:attribute type="xs:string" name="unit"
    use="required"/>
243                                </xs:extension>
244                              </xs:simpleContent>
245                            </xs:complexType>
246                          </xs:element>
247                          <xs:element name="macros">
248                            <xs:complexType>
249                              <xs:sequence>
250                                <xs:element name="protein">
251                                  <xs:complexType>
252                                    <xs:simpleContent>
253                                      <xs:extension base="xs:string">
254                                        <xs:attribute type="xs:string" name="unit"
    use="required"/>
255                                      </xs:extension>
256                                    </xs:simpleContent>
257                                  </xs:complexType>
258                                </xs:element>
259                                <xs:element name="fat">
260                                  <xs:complexType>
261                                    <xs:simpleContent>
262                                      <xs:extension base="xs:string">
263                                        <xs:attribute type="xs:string" name="unit"
    use="required"/>
264                                      </xs:extension>
265                                    </xs:simpleContent>
266                                  </xs:complexType>
267                                </xs:element>
268                                <xs:element name="carbs">
269                                  <xs:complexType>
270                                    <xs:simpleContent>
271                                      <xs:extension base="xs:string">
272                                        <xs:attribute type="xs:string" name="unit"
    use="required"/>
273                                      </xs:extension>
274                                    </xs:simpleContent>
275                                  </xs:complexType>
```

localhost:4649/?mode=xml 5/7



5/27/2021 Restau.xsd

```
276                                </xs:element>
277                              </xs:sequence>
278                            </xs:complexType>
279                          </xs:element>
280                        </xs:sequence>
281                        <xs:attribute type="xs:string" name="id" use="required"/>
282                      </xs:complexType>
283                    </xs:element>
284                  </xs:sequence>
285                </xs:complexType>
286              </xs:element>
287            </xs:sequence>
288          </xs:complexType>
289        </xs:element>
290        <xs:element name="beverage">
291          <xs:complexType>
292            <xs:sequence>
293              <xs:element name="diet_drinks">
294                <xs:complexType>
295                  <xs:sequence>
296                    <xs:element name="diet_drink" maxOccurs="unbounded"
    minOccurs="2">
297                      <xs:complexType>
298                        <xs:sequence>
299                          <xs:element type="xs:string" name="name"/>
300                          <xs:element name="price">
301                            <xs:complexType>
302                              <xs:simpleContent>
303                                <xs:extension base="xs:string">
304                                  <xs:attribute type="xs:string" name="currency"
    use="required"/>
305                                </xs:extension>
306                              </xs:simpleContent>
307                            </xs:complexType>
308                          </xs:element>
309                          <xs:element name="calories">
310                            <xs:complexType>
311                              <xs:simpleContent>
312                                <xs:extension base="xs:string">
313                                  <xs:attribute type="xs:string" name="unit"
    use="required"/>
314                                </xs:extension>
315                              </xs:simpleContent>
316                            </xs:complexType>
317                          </xs:element>
318                          <xs:element name="volume">
319                            <xs:complexType>
320                              <xs:simpleContent>
321                                <xs:extension base="xs:string">
322                                  <xs:attribute type="xs:string" name="unit"
    use="required"/>
323                                </xs:extension>
324                              </xs:simpleContent>
325                            </xs:complexType>
326                          </xs:element>
327                        </xs:sequence>
328                        <xs:attribute type="xs:string" name="id" use="required"/>
329                      </xs:complexType>
330                    </xs:element>
331                  </xs:sequence>
```

localhost:4649/?mode=xml 6/7

5/27/2021 Restau.xsd

```
332                </xs:complexType>
333              </xs:element>
334            </xs:sequence>
335          </xs:complexType>
336        </xs:element>
337      </xs:sequence>
```

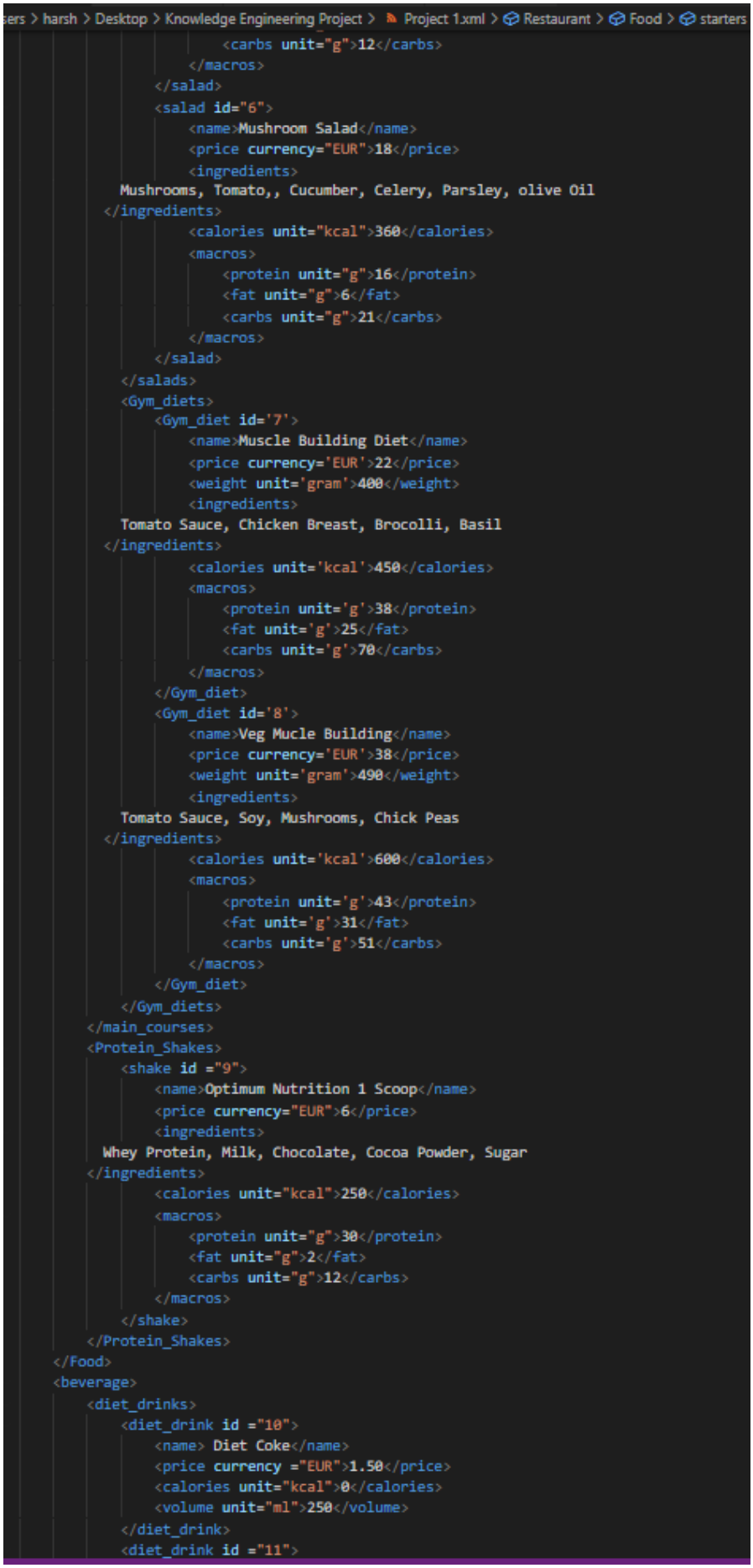
(c) Create two example XML documents that are valid against your XSD. Give four supplementary partial examples of how your XSD would catch invalid pieces of XML documents.

[10 Marks]

Below is the first example for my XML

XML NUMBER – 1

```
<Restaurant>
    <Food>
        <starters>
            <starter id ="1">
                <name>Pav Bhaji</name>
                <price currency ="EUR">18</price>
                <ingredients>
          Bhaji, Pav Bread, Herbs
        </ingredients>
                <calories unit="kcal">250</calories>
                <macros>
                    <protein unit="g">5.9</protein>
                    <fat unit="g">12</fat>
                    <carbs unit="g">20</carbs>
                </macros>
            </starter>
            <starter id ="2">
                <name> Mexican Burrito</name>
                <price currency ="EUR">10</price>
                <ingredients>
          Whole Wheat Tortilla, Rice, Chicken, Gheese, Sweet Chilli Sauce
        </ingredients>
                <calories>700</calories>
                <macros>
                    <protein unit="g">18</protein>
                    <fat unit="g">14</fat>
                    <carbs unit="g">25</carbs>
                </macros>
            </starter>
            <starter id ="3">
                <name>Cheese Burger</name>
                <price currency ="EUR">13.50</price>
                <ingredients>
          Burger Bun, Cheese, Patty, Herbs
        </ingredients>
                <calories unit="kcal">450</calories>
                <macros>
                    <protein unit="g">10.9</protein>
                    <fat unit="g">5</fat>
                    <carbs unit="g">20</carbs>
                </macros>
            </starter>
        </starters>
        <main_courses>
            <salads>
                <salad id="4">
                    <name>Basic Salad</name>
                    <price currency="EUR">22</price>
                    <ingredients>
          Tomatoes, Spinach , Basil Leaves, Cucumber, Carrot
        </ingredients>
                    <calories unit="kcal">380</calories>
                    <macros>
                        <protein unit="g">2</protein>
                        <fat unit="g">9</fat>
                        <carbs unit="g">29</carbs>
                    </macros>
                </salad>
                <salad id = "5">
```


sers > harsh > Desktop > Knowledge Engineering Project > Project 1.xml > Restaurant > Food > starters
<carbs unit="g">12</carbs>
</macros>
</salad>
<salad id="6">
<name>Mushroom Salad</name>
<price currency="EUR">18</price>
<ingredients>
Mushrooms, Tomato,, Cucumber, Celery, Parsley, olive Oil
</ingredients>
<calories unit="kcal">360</calories>
<macros>
<protein unit="g">16</protein>
<fat unit="g">6</fat>
<carbs unit="g">21</carbs>
</macros>
</salad>
</salads>
<Gym_diets>
<Gym_diet id='7'>
<name>Muscle Building Diet</name>
<price currency='EUR'>22</price>
<weight unit='gram'>400</weight>
<ingredients>
Tomato Sauce, Chicken Breast, Brocolli, Basil
</ingredients>
<calories unit='kcal'>450</calories>
<macros>
<protein unit='g'>38</protein>
<fat unit='g'>25</fat>
<carbs unit='g'>70</carbs>
</macros>
</Gym_diet>
<Gym_diet id='8'>
<name>Veg Mucle Building</name>
<price currency='EUR'>38</price>
<weight unit='gram'>490</weight>
<ingredients>
Tomato Sauce, Soy, Mushrooms, Chick Peas
</ingredients>
<calories unit='kcal'>600</calories>
<macros>
<protein unit='g'>43</protein>
<fat unit='g'>31</fat>
<carbs unit='g'>51</carbs>
</macros>
</Gym_diet>
</Gym_diets>
</main_courses>
<Protein_Shakes>
<shake id ="9">
<name>Optimum Nutrition 1 Scoop</name>
<price currency="EUR">6</price>
<ingredients>
Whey Protein, Milk, Chocolate, Cocoa Powder, Sugar
</ingredients>
<calories unit="kcal">250</calories>
<macros>
<protein unit="g">30</protein>
<fat unit="g">2</fat>
<carbs unit="g">12</carbs>
</macros>
</shake>
</Protein_Shakes>
</Food>
<beverage>
<diet_drinks>
<diet_drink id ="10">
<name> Diet Coke</name>
<price currency ="EUR">1.50</price>
<calories unit="kcal">0</calories>
<volume unit="ml">250</volume>
</diet_drink>
<diet_drink id ="11">

```
        <diet_drinks>
            <diet_drink id ="10">
                <name> Diet Coke</name>
                <price currency ="EUR">1.50</price>
                <calories unit="kcal">0</calories>
                <volume unit="ml">250</volume>
            </diet_drink>
            <diet_drink id ="11">
                <name>Green Tea</name>
                <price currency ="EUR">2.50</price>
                <calories unit="kcal">20</calories>
                <volume unit="ml">250</volume>
            </diet_drink>
        </diet_drinks>
    </beverage>
</Restaurant>
```

This is the Second Example in which I have added various elements to the starter, and what I observed was that when I am violating minimum occurrence, the XML doesn't validate

XML Number -2

```
<Restaurant>
    <Food>
        <starters>
            <starter id ="1">
                <name>Chowmein ROll</name>
                <price currency ="EUR">5</price>
                <ingredients>
          Chowmein
        </ingredients>
                <calories unit="kcal">240</calories>
                <macros>
                    <protein unit="g">9.9</protein>
                    <fat unit="g">15</fat>
                    <carbs unit="g">40</carbs>
                </macros>
            </starter>
            <starter id ="2">
                <name> Mexican Burrito</name>
                <price currency ="EUR">10</price>
                <ingredients>
          Whole Wheat Tortilla, Rice, Chicken, Gheese, Sweet Chilli Sauce
        </ingredients>
                <calories>700</calories>
                <macros>
                    <protein unit="g">18</protein>
                    <fat unit="g">14</fat>
                    <carbs unit="g">25</carbs>
                </macros>
            </starter>
            <starter id ="12">
                <name> Afghani Chaap</name>
                <price currency ="EUR">50</price>
                <ingredients>
          Soyabean Chaap, Cream
        </ingredients>
                <calories>700</calories>
                <macros>
                    <protein unit="g">18</protein>
                    <fat unit="g">14</fat>
                    <carbs unit="g">25</carbs>
                </macros>
            </starter>
            <starter id ="3">
                <name>chilli Cheese Burger</name>
                <price currency ="EUR">15.50</price>
                <ingredients>
          Burger Bun, Cheese, Patty, Herbs
        </ingredients>
                <calories unit="kcal">450</calories>
                <macros>
                    <protein unit="g">10.9</protein>
                    <fat unit="g">5</fat>
                    <carbs unit="g">20</carbs>
                </macros>
            </starter>
        </starters>
        <main_courses>
            <salads>

                <salad id="12">
                    <name>Deli Style Salad</name>
                    <price currency="EUR">32</price>
                    <ingredients>
            Salad, Burger
          </ingredients>
                    <calories unit="kcal">200</calories>
                    <macros>
                        <protein unit="g">2</protein>
                        <fat unit="g">9</fat>
                        <carbs unit="g">29</carbs>
                    </macros>
                </salad>
                <salad id="4">
```

```
            </salad>
            <salad id="4">
                <name>Basic Salad</name>
                <price currency="EUR">22</price>
                <ingredients>
      Tomatoes, Spinach , Basil Leaves, Cucumber, Carrot
    </ingredients>
                <calories unit="kcal">380</calories>
                <macros>
                    <protein unit="g">2</protein>
                    <fat unit="g">9</fat>
                    <carbs unit="g">29</carbs>
                </macros>
            </salad>
            <salad id = "5">
                <name>Beetroot Chicken Salad</name>
                <price currency="EUR">17</price>
                <ingredients>
      Beetroot, Tomato, Cucumber, Carrots, Chicken
    </ingredients>
                <calories unit="kcal">255</calories>
                <macros>
                    <protein unit="g">25</protein>
                    <fat unit="g">8</fat>
                    <carbs unit="g">12</carbs>
                </macros>
            </salad>
            <salad id="6">
                <name>Mushroom Salad</name>
                <price currency="EUR">18</price>
                <ingredients>
      Mushrooms, Tomato,, Cucumber, Celery, Parsley, olive Oil
    </ingredients>
                <calories unit="kcal">360</calories>
                <macros>
                    <protein unit="g">16</protein>
                    <fat unit="g">6</fat>
                    <carbs unit="g">21</carbs>
                </macros>
            </salad>
        </salads>
        <Gym_diets>
            <Gym_diet id='7'>
                <name>Muscle Building Diet</name>
                <price currency='EUR'>22</price>
                <weight unit='gram'>400</weight>
                <ingredients>
      Tomato Sauce, Chicken Breast, Brocolli, Basil
    </ingredients>
                <calories unit='kcal'>450</calories>
                <macros>
                    <protein unit='g'>38</protein>
                    <fat unit='g'>25</fat>
                    <carbs unit='g'>70</carbs>
                </macros>
            </Gym_diet>
            <Gym_diet id='8'>
                <name>Veg Mucle Building</name>
                <price currency='EUR'>38</price>
                <weight unit='gram'>490</weight>
                <ingredients>
      Tomato Sauce, Soy, Mushrooms, Chick Peas
    </ingredients>
                <calories unit='kcal'>600</calories>
                <macros>
                    <protein unit='g'>43</protein>
                    <fat unit='g'>31</fat>
                    <carbs unit='g'>51</carbs>
                </macros>
            </Gym_diet>
        </Gym_diets>
    </main_courses>
```

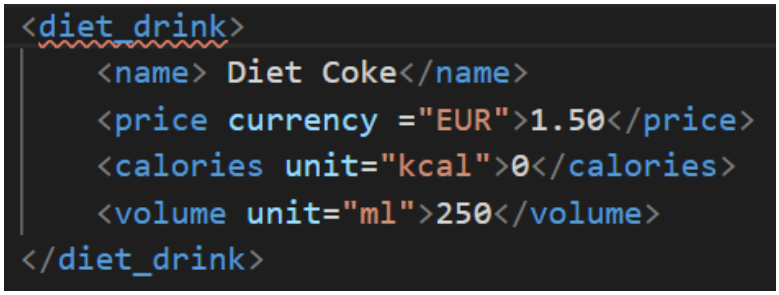
```
                    <fat unit='g'>31</fat>
                    <carbs unit='g'>51</carbs>
                </macros>
            </Gym_diet>
        </Gym_diets>
    </main_courses>
    <Protein_Shakes>
        <shake id ="9">
            <name>Optimum Nutrition 1 Scoop</name>
            <price currency="EUR">6</price>
            <ingredients>
      Whey Protein, Milk, Chocolate, Cocoa Powder, Sugar
    </ingredients>
            <calories unit="kcal">250</calories>
            <macros>
                <protein unit="g">30</protein>
                <fat unit="g">2</fat>
                <carbs unit="g">12</carbs>
            </macros>
        </shake>

        <shake id ="25">
            <name>Dymatize 1 Scoop</name>
            <price currency="EUR">6</price>
            <ingredients>
      Whey Protein, Milk, Chocolate, Cocoa Powder, Sugar
    </ingredients>
            <calories unit="kcal">250</calories>
            <macros>
                <protein unit="g">30</protein>
                <fat unit="g">2</fat>
                <carbs unit="g">12</carbs>
            </macros>
        </shake>
    </Protein_Shakes>
</Food>
<beverage>
    <diet_drinks>
        <diet_drink id ="10">
            <name> Diet Coke</name>
            <price currency ="EUR">1.50</price>
            <calories unit="kcal">0</calories>
            <volume unit="ml">250</volume>
        </diet_drink>
        <diet_drink id ="11">
            <name>Green Tea</name>
            <price currency ="EUR">2.50</price>
            <calories unit="kcal">20</calories>
            <volume unit="ml">250</volume>
        </diet_drink>
        <diet_drink id ="19">
            <name>Ice Tea</name>
            <price currency ="EUR">7.50</price>
            <calories unit="kcal">200</calories>
            <volume unit="ml">250</volume>
        </diet_drink>
        <diet_drink id ="29">
            <name>Hot Tea</name>
            <price currency ="EUR">1.50</price>
            <calories unit="kcal">200</calories>
            <volume unit="ml">250</volume>
        </diet_drink>

    </diet_drinks>
</beverage>
</Restaurant>
```

Partial Examples of XSD catching invalid pieces of XML documents

I)

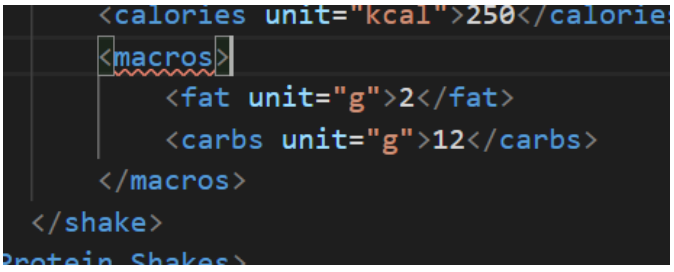
The minimum occurrence of starters has been set as 3 if we add only 2 starters in the XML document it will show an error. The error would be like" The content of element 'starters' is not complete. 'starter' is expected to occur a minimum of '3' times. One more instance is required to satisfy this constraint."

II)



If the Id from diet_drinks is deleted, it will show an error, and the error would be Attribute 'id' must appear on element 'diet_drink.'

III)



The macros has protein, fats , carbs as the required attributes and the error would be nvalid content was found starting with element 'fat'. One of '{protein}' is expected.

IV)

```
<shake id ="9">
    <name>Optimum Nutrition 1 Scoop</name>
    <ingredients>
Whey Protein, Milk, Chocolate, Cocoa Powder, Sugar
Ingredients>
    <calories unit="kcal">250</calories>
```

Here since price is missing therefore it will show an error and the error would be like Invalid content was found starting with element 'ingredients'. One of '{price}' is expected.

(d) Demonstrate, using four XPATH expressions on one of your XML documents, different features of the XPATH specification.

[10 Marks]

The Query and the output for XPATH is Shown Below

QUERY No. 1 : /Restaurant/Food/main_courses/Gym_diets/Gym_diet[price<25]/calories

```
Element='<calories unit="kcal">450</calories>'
```

QUERY No. 2 //diet_drinks/diet_drink

```
Element='<diet_drink id="10">
                        <name> Diet Coke</name>
                        <price currency="EUR">1.50</price>
                        <calories unit="kcal">0</calories>
                        <volume unit="ml">250</volume>
                    </diet_drink>'
Element='<diet_drink id="11">
                        <name>Green Tea</name>
                        <price currency="EUR">2.50</price>
                        <calories unit="kcal">20</calories>
                        <volume unit="ml">250</volume>
                    </diet_drink>'
```

Query No. 3 //name

```
Element='<name>Pav Bhaji</name>'
Element='<name> Mexican Burrito</name>'
Element='<name>Cheese Burger</name>'
Element='<name>Basic Salad</name>'
Element='<name>Beetroot Chicken Salad</name>'
Element='<name>Mushroom Salad</name>'
Element='<name>Muscle Building Diet</name>'
Element='<name>Veg Mucle Building</name>'
Element='<name>Optimum Nutrition 1 Scoop</name>'
Element='<name> Diet Coke</name>'
Element='<name>Green Tea</name>'
```

Query No. 4 //diet_drink[last()]

```
Element='<diet_drink id="11">
                                <name>Green Tea</name>
                                <price currency="EUR">2.50</price>
                                <calories unit="kcal">20</calories>
                                <volume unit="ml">250</volume>
                        </diet_drink>'
```

(e) Create an XSLT that will transform documents that correspond to your XSD into HTML. Provide a step-by-step illustration of how this XSLT could be used to process one of your example XML documents.

[10 Marks]

The XSLT is used to get a HTML file and the procedure I followed is –

Step 1 : Use of <xsl:template match="/Restaurant"> this defines the template for the entire document.

Step 2 : Use of HTML tags to define the structure of the webpage.

Step 3 : I have created Tables for starters, main course gymdiets , protein shake and beverage.

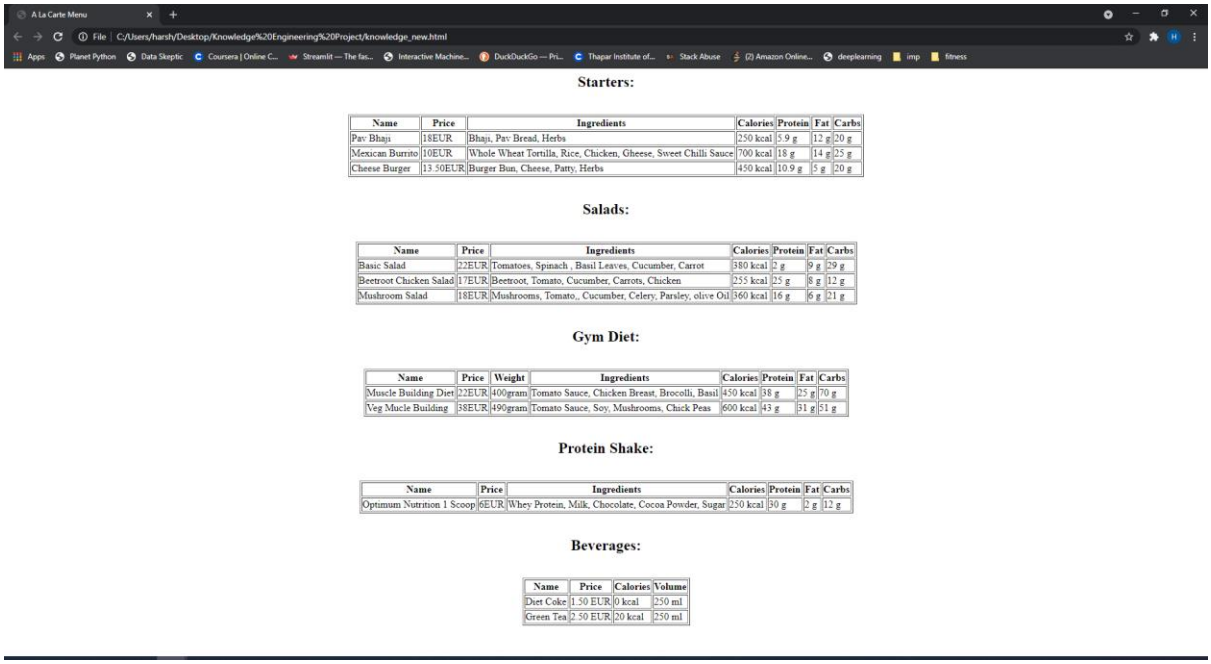
Step 4 : input of data into table via <xsl:for-each> tag and all the data is being input via this tag, I use this tag to input into each table separately

Step 5: After the XSl style sheet is made its being linked to XML and we can get HTML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
	<xsl:template match="/Restaurant">
	<html>
			<head>
				<meta charset="utf-8"/>
				<title>A La Carte Menu</title>
			</head>
			<body>
		<center>
		    <h2>Starters:</h2>
		    <br></br>
		    <table border="1">
		        <tr>
		            <th>Name</th>
                                        <th>Price</th>
                                        <th>Ingredients</th>
                                        <th>Calories</th>
                                        <th>Protein</th>
                                        <th>Fat</th>
                                        <th>Carbs</th>

		        </tr>
		        <xsl:for-each select="/Restaurant/Food/starters/starter">
		            <tr>
                                                <td ><xsl:value-of select="name"/></td>
                                                <td><xsl:value-of select="price"/>EUR</td>
                                                <td><xsl:value-of select="ingredients"/></td>
                                                <td><xsl:value-of select="calories"/> kcal</td>
                                                <td><xsl:value-of select="macros/protein" /> g</td>
                                                <td><xsl:value-of select="macros/fat" /> g</td>
                                                <td><xsl:value-of select="macros/carbs" /> g</td>
		            </tr>
		        </xsl:for-each>

		    </table>
		    <br></br>

		    <h2>Salads:</h2>
		    <br></br>
		    <table border="1">
		        <tr>
```

```
		    <br></br>

		    <h2>Salads:</h2>
		    <br></br>
		    <table border="1">
		        <tr>
		            <th>Name</th>
                                        <th>Price</th>
                                        <th>Ingredients</th>
                                        <th>Calories</th>
                                        <th>Protein</th>
                                        <th>Fat</th>
                                        <th>Carbs</th>
		        </tr>

		        <xsl:for-each select="/Restaurant/Food/main_courses/salads/salad">
		            <tr>
		                <td ><xsl:value-of select="name"/></td>
		                <td><xsl:value-of select="price"/>EUR</td>
		                <td><xsl:value-of select="ingredients"/></td>
		                <td><xsl:value-of select="calories"/> kcal</td>
		                <td><xsl:value-of select="macros/protein" /> g</td>
		                <td><xsl:value-of select="macros/fat" /> g</td>
		                <td><xsl:value-of select="macros/carbs" /> g</td>
		            </tr>
		        </xsl:for-each>
		    </table>
		    <br></br>

		    <h2>Gym Diet:</h2>
		    <br></br>
		    <table border="1">
		        <tr>
		            <th>Name</th>
                                        <th>Price</th>
		            <th>Weight</th>
                                        <th>Ingredients</th>
                                        <th>Calories</th>
                                        <th>Protein</th>
                                        <th>Fat</th>
                                        <th>Carbs</th>
		        </tr>
```

```
        <xsl:for-each select="/Restaurant/Food/main_courses/Gym_diets/Gym_diet">
            <tr>
                <td ><xsl:value-of select="name"/></td>
                <td><xsl:value-of select="price"/>EUR</td>
                <td><xsl:value-of select="weight"/>gram</td>
                <td><xsl:value-of select="ingredients"/></td>
                <td><xsl:value-of select="calories"/> kcal</td>
                <td><xsl:value-of select="macros/protein" /> g</td>
                <td><xsl:value-of select="macros/fat" /> g</td>
                <td><xsl:value-of select="macros/carbs" /> g</td>
            </tr>
        </xsl:for-each>
    </table>
    <br></br>

    <h2>Protein Shake:</h2>
    <br></br>
    <table border="1">
        <tr>
            <th>Name</th>
                                        <th>Price</th>
                                        <th>Ingredients</th>
                                        <th>Calories</th>
                                        <th>Protein</th>
                                        <th>Fat</th>
                                        <th>Carbs</th>
        </tr>

        <xsl:for-each select="/Restaurant/Food/Protein_Shakes/shake">
            <tr>
                <td ><xsl:value-of select="name"/></td>
                <td><xsl:value-of select="price"/>EUR</td>
                <td><xsl:value-of select="ingredients"/></td>
                <td><xsl:value-of select="calories"/> kcal</td>
                <td><xsl:value-of select="macros/protein" /> g</td>
                <td><xsl:value-of select="macros/fat" /> g</td>
                <td><xsl:value-of select="macros/carbs" /> g</td>
            </tr>
        </xsl:for-each>

    </table>
```



Starters:

Name	Price	Ingredients	Calories	Protein	Fat	Carbs
Pav Bhaji	18EUR	Bhaji, Pav Bread, Herbs	250 kcal	5.9 g	12 g	20 g
Mexican Burrito	10EUR	Whole Wheat Tortilla, Rice, Chicken, Gheese, Sweet Chilli Sauce	700 kcal	18 g	14 g	25 g
Cheese Burger	13.50EUR	Burger Bun, Cheese, Patty, Herbs	450 kcal	10.9 g	5 g	20 g

Salads:

Name	Price	Ingredients	Calories	Protein	Fat	Carbs
Basic Salad	22EUR	Tomatoes, Spinach , Basil Leaves, Cucumber, Carrot	380 kcal	2 g	9 g	29 g
Beetroot Chicken Salad	17EUR	Beetroot, Tomato, Cucumber, Carrots, Chicken	255 kcal	25 g	8 g	12 g
Mushroom Salad	18EUR	Mushrooms, Tomato,, Cucumber, Celery, Parsley, olive Oil	360 kcal	16 g	6 g	21 g

Gym Diet:

Name	Price	Weight	Ingredients	Calories	Protein	Fat	Carbs
Muscle Building Diet	22EUR	400gram	Tomato Sauce, Chicken Breast, Brocolli, Basil	450 kcal	38 g	25 g	70 g
Veg Mucle Building	38EUR	490gram	Tomato Sauce, Soy, Mushrooms, Chick Peas	600 kcal	43 g	31 g	51 g

Protein Shake:

Name	Price	Ingredients	Calories	Protein	Fat	Carbs
Optimum Nutrition 1 Scoop	6EUR	Whey Protein, Milk, Chocolate, Cocoa Powder, Sugar	250 kcal	30 g	2 g	12 g

Beverages:

Name	Price	Calories	Volume
Diet Coke	1.50 EUR	0 kcal	250 ml
Green Tea	2.50 EUR	20 kcal	250 ml

(f) Discuss how, with partial examples, XSLT may be used to transform documents that correspond to your XSD into RDF or RDFS.

[5 Marks]

For transformation of XSLT to RDF

URI's of the resources should be created, and they should be connected to the RDF triple structure. [12] this paper introduces us with an XML to RDF transformation approach that is based upon the mapping of RDF triple templates

Expected transformation –

```
<price currency="EUR">6</price>
```

It could be

1 <file:///STELLAI_3.rdf#RTML /Restaurant/Food/starters/starter/price>

2 nsl:Currency "EUR";

3 rdf:value "6".

Q2 (2000 word limit)

Web search, using tools such as Google and Bing, has been the dominant means of discovering information online, but semantic-based approaches are becoming increasingly important in making sense of such information. Discuss this statement in reference to the emergence of the Semantic Web and its evolving interaction with web search. What are the current technical limitations of how the Semantic Web is realised that restrict semantic-based approaches? In your opinion how will these technologies (and related ideologies) impact how we discover, access and interact with content online over the next 10 years.

[50 Marks]

The worldwide Web (www), also known as the Web, is an information space and has massive amounts of organized and unorganized information. The web resources are identified by URI's interlinked by hypertext links and can be accessed via the internet. The Web made its roots merely 30 years back, i.e., around 1992, from 1993 -1996 there was a rapid growth in the Web i.e., from 130 sites to 600,000 sites and its evergrowing from then, there are around 1.7 billion websites. Search engines such as Google, Bing are designed to search and extract the required information from this humungous pool of data on the Web.

According to a survey dated April 2021, Google has 92.24% of the search engine market share worldwide. Google uses an algorithm called PageRank [1]; the search engine assigns a rank to each page in its index upon several ranking factors. It is a very complex algorithm. Google searches in billions of web pages on various factors such as query, relevance, and location. Google uses anchor test and HITS(Hypertext – Induced Topic Search) strategy. Google's algorithm focuses on displaying the top web pages according to the keyword that the user has searched for. This search has a special term called 'focalized search' – no ability to explore data and it assumes that the user knows everything about the query and know exact terms to investigate, this is good in case of basic search mechanism but if one doesn't know at all what to search it's a drawback[3]. We often have to go through various ineffectual links (not relevant) in the search results despite all this. It can be concluded that there are multiple drawbacks of the exiting means/algorithm for searching. Also, the Web was designed with the goal that it would be beneficial for humans as well as machines. The existing Web 2.0 was a data explosion; it introduced us to crowdsourcing, social networks, and digital media. The traditional Web has various drawbacks such as accessing the data; users search for data on the Web asking "Which document contain these phrases" therefore, we get the search results according to the vocabulary we get single web pages since documents are indexed, current paradigm relies on returning the best fit document for the search query. This is where Semantic Web can be a game-changer; it can extend the Web with meaningful data.

The third generation web: the Web of data or so-called semantic Web is an upgrade from the Web of documents; here, the data is in the center of processing. The web is seen as a vast decentralized base or knowledge base of machine-accessible data. This was first proposed in the late 1990's by Tim Berner Lee's, and he stated that "The Web of human-readable document is being merged with a web of machine-understandable data. The potential of the mixture of humans and machines working together and communication through the Web could be immense" [2]. So, in short, the semantic Web would help computers make meaningful interpretations similar to how humans interpret information. Semantic Web envisages Web around machine-understandable content. It hopes to empower computers to analyze, reason, and make predictions of the data confidently. It will also enable computers to communicate meaningful content to humans as well as to each other. The traditional AI faces a lot of problems in extracting information from the Web because, at present, the data distributed throughout the Web is highly heterogenous and unstructured; the Semantic Web is the logical next frontier in this regard. It can organize and link data over the Web in a consistent manner which would allow "us to

pose queries rather than browse documents, to infer new knowledge from existing facts, and to identify inconsistencies" [4].

The Semantic Web Technology stack includes RDF (Resource Description Framework). RDF is the building block of the semantic Web on the information exchange layer. Everything we encode in RDF will be serialized in XML (extensible Markup Language). Above the information exchange layer is the layer where we can build models with web ontology language or RDFS; for these models, we can create a logical rule, and we can apply logical frameworks on top of that models. OWL defines the types of relationships expressed in RDF using an XML vocabulary to indicate the hierarchies and relationships between different resources. This is the very definition of "ontology" in the Semantic Web context: a schema that formally defines the hierarchies and relationships between other resources. Semantic Web ontologies consist of a taxonomy and a set of inference rules from which machines can make logical conclusions. . Semantic Web revolves around linked data; for achieving the goal of an efficient Semantic web, linked data is the key. Data can be qualified as Linked data if it possesses uniqueness, loop-up ability, query-able nature, and can link pre-existing URI's. In short, linked data can be seen as a technique for publishing and consuming structured data on the Web

As of now, the Semantic Web is in its nascent stages; there is a Linked Open Data Cloud which is publically available RDF data on the Web, the Web of data as of August 2015 has 9960 datasets, more than 85 billion facts and 800 million links A joint effort made by Google, Microsoft, Yahoo, and Yandex proposed a solution called Schema.org to promote structured data in web pages with a common vocabulary. This structured data can help provide richer information in the search results[6]. Semantic technology is in use without even the knowledge of people; the users are using more natural speech for searches these days. Companies are trying to upgrade their search engines so that they should "focus less on keywords and more on intent-based collective intelligence" [a]. Google's newly developed algorithm "Hummingbird" allows user to conduct "conversational searches," which means it will focus on the entire sentence, not just keywords

Moving from where we are now to full semantic would take a lot of time, which won't happen overnight. But at present, it faces significant issues. Combining information from multiple sources on the semantic Web can be very problematic because different sources may use different ontologies, i.e. different vocabularies for the same thing. Different sources may use different6 semantics. It may contain different degrees of reliability. Therefore it's necessary to resolve logical contradictions. At present, RDF defines a way to solve possible contradictions at the cost of computing power.

Moreover there is a lack of trust and credibility in information sources because of a high volume of malicious information present on the Web. The current success of semantic technologies lies in combining traditional information retrieval approaches, soft AI, and many statistics. We need to overcome these challenges for achieving vision of Semantic Web. The future of the Web is still unknown to all, at present, the technology for many applications is still not mature

In my opinion, if we could achieve semantic Web over the next 10 years, it would change the way how we will discover, access, and interact with content online. It would not only change the course of searching information but would also change the control of information; for example every website would be a graph of inter-related topics, and therefore there would be no focus spot where data is stored and hence no single entity to control the data. This would improve privacy and data control over one's data. The web 3.0 (Semantic Web) would act as a personal assistant who would practically know everything with answers to public information stored on the internet (AI would help search engines to show personalized search results). It will use the information present on the Web to make connections, unlike web 2.0, where people connect using the internet. With the future web we can also search a complex query "I want to go to London on holidays for a week stay in a budget hotel and visit famous places there" The web search tool will do all the work and would give you organized results based on your query. This will not only shape the future but would also launch an age of knowledge, and in the next ten years, it will take us from the age of information to age of knowledge. Moreover there would be various advancements in terms of voice recognition field and would revolutionize assistants like

SIRI, ALEXA. "Within the next ten years, the semantic web will take us from the age of information to the age of knowledge. Simple tools and services will allow individuals, corporations, and governments to quickly glean meaning from the vast amounts of data they have compiled. This move from a 'World Wide Web' to a 'worldwide database' will allow for hidden relationships and connections to surface quickly, driving both innovation and (unfortunately) exploitation. The impact of the semantic web will be substantial. It will help create new industries, influence campaign strategies and lead to ground-breaking discoveries in both science and medicine." –**Bryan Trogdon**, president of First Semantic.[10]. Many technologies would also revolutionize after the development of Semantic Web. It would help in advancement of NLP (Natural Language Processing), AR (Augmented Reality) and would change the way people look through the web

Semantic Blockchain technology would help in revolutionizing web 3.0. It would enable us to interconnect and store data smartly and securely, e.g. smart contracts. With the help of a decentralized ledger in the form of dApps(Decentralized Apps), apps would run independently without a central server. It would also be possible to easily connect with other dApps and track specific information[9].

The Semantic Web is the future, and it will free humans from the most time-consuming tasks we face today, Autonomous systems that work on our behalf and perform the task of organizing, retrieving, and making sense of the information present on the Web, will enable us to delve into the next level of cognitive reasoningand this is the next step to the path of human understanding. "The level of advancement of a civilization is measured by the complexity of the tasks that it automates and frees from the day-to-day attention of its members." [11]

References

[1] https://en.wikipedia.org/wiki/PageRank

[2] https://e-envir.sciencesconf.org/data/pages/J3_CM7_WebSem_ZiebelinDanielle.pdf

[3] https://community.aiim.org/blogs/johannes-scholtes/2010/06/24/understand-the-benefits-and-limitations-of-traditional-web-search-engines-such-as-google-when-you-use-their-appliances-and-technology-in-house-for-mission-critical-applications

[4] (PDF) The Evolution of the Semantic Web (researchgate.net)

[5] https://livebook.manning.com/book/linked-data/chapter-1/188

[6] https://stackoverflow.blog/2020/12/10/the-semantic-future-of-the-web/?cb=1

[7] https://wiki.p2pfoundation.net/Semantic_Web

[a] https://www.wired.com/insights/2014/02/search-today-beyond-optimizing-semantic-web/

[9] https://wiredelta.com/web-3-0-change-internet-future/

[10] Prospects for the semantic web | Pew Research Center. https://www.pewresearch.org/internet/2010/05/04/prospects-for-the-semantic-web/

[11] https://www.cse.iitb.ac.in/~pararth09/semantic/SemanticWeb_seminar.pptx

[12] https://dl.acm.org/doi/10.1145/2814864.2814880