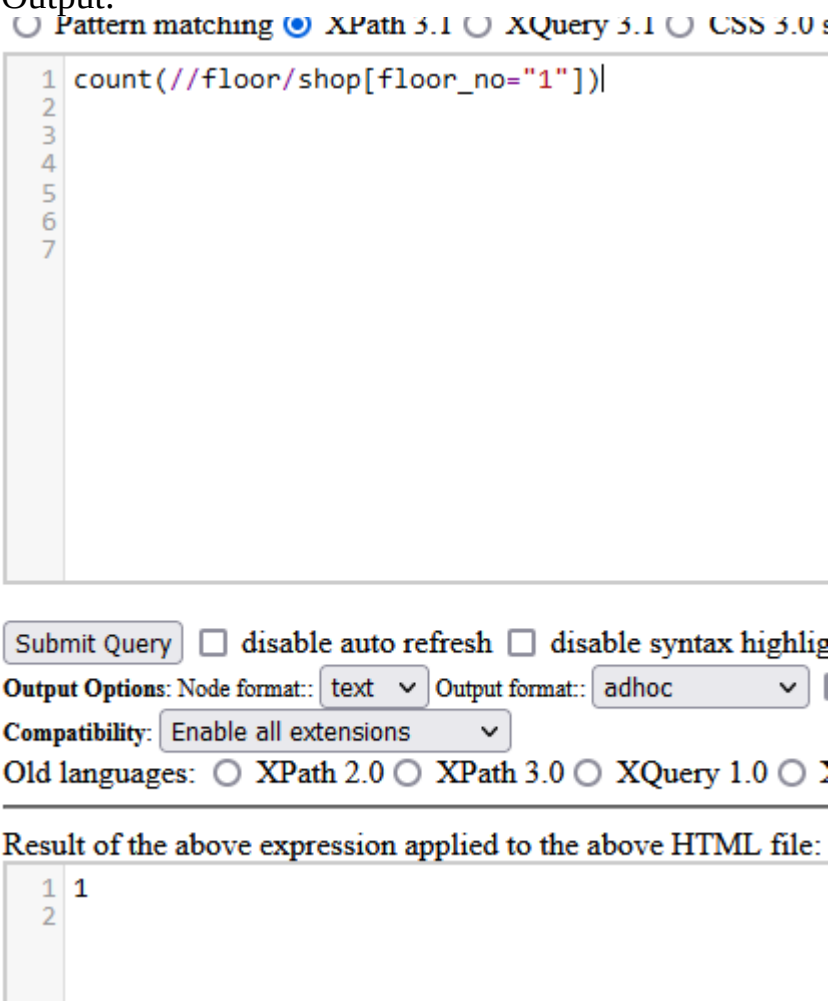
ASSIGNMENT - 1

Roll No. = CS22M055
NAME = MAHENDRA LACHETA.

XPath:

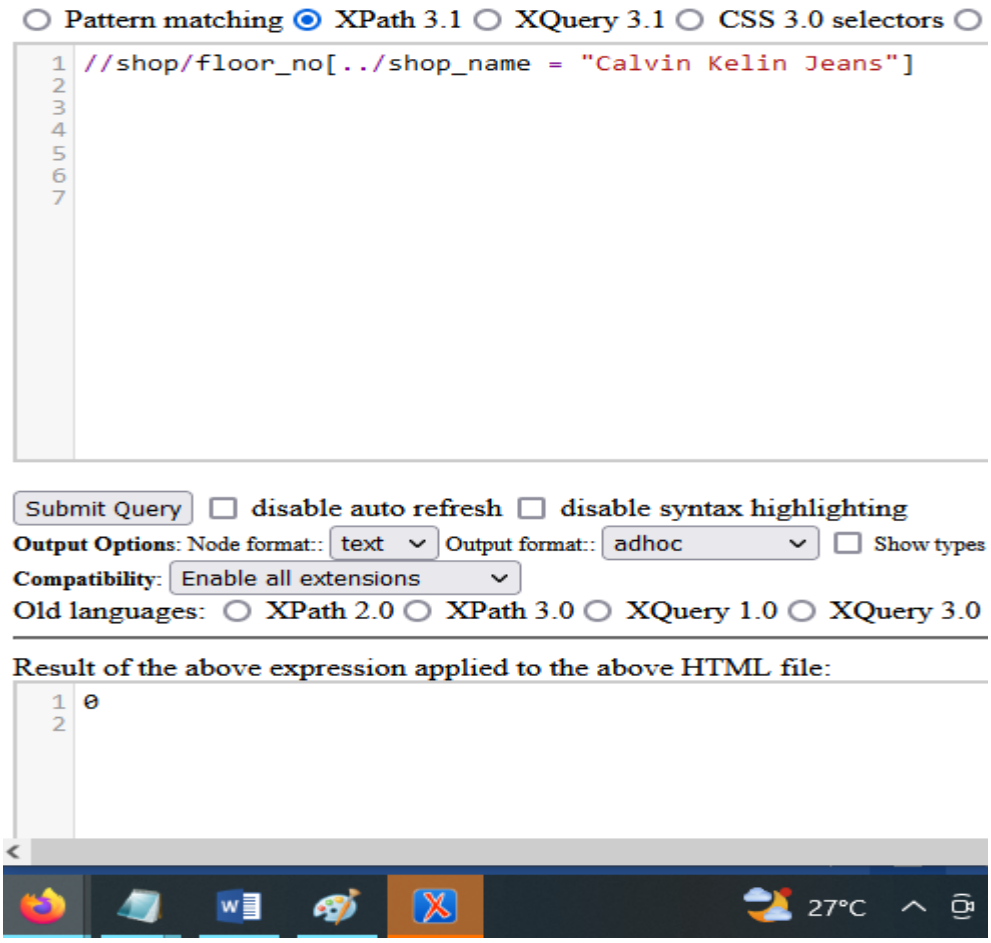
1. Count the number of shops no floor_no is 1.
   count(//floor/shop[floor_no="1"])

   Output:



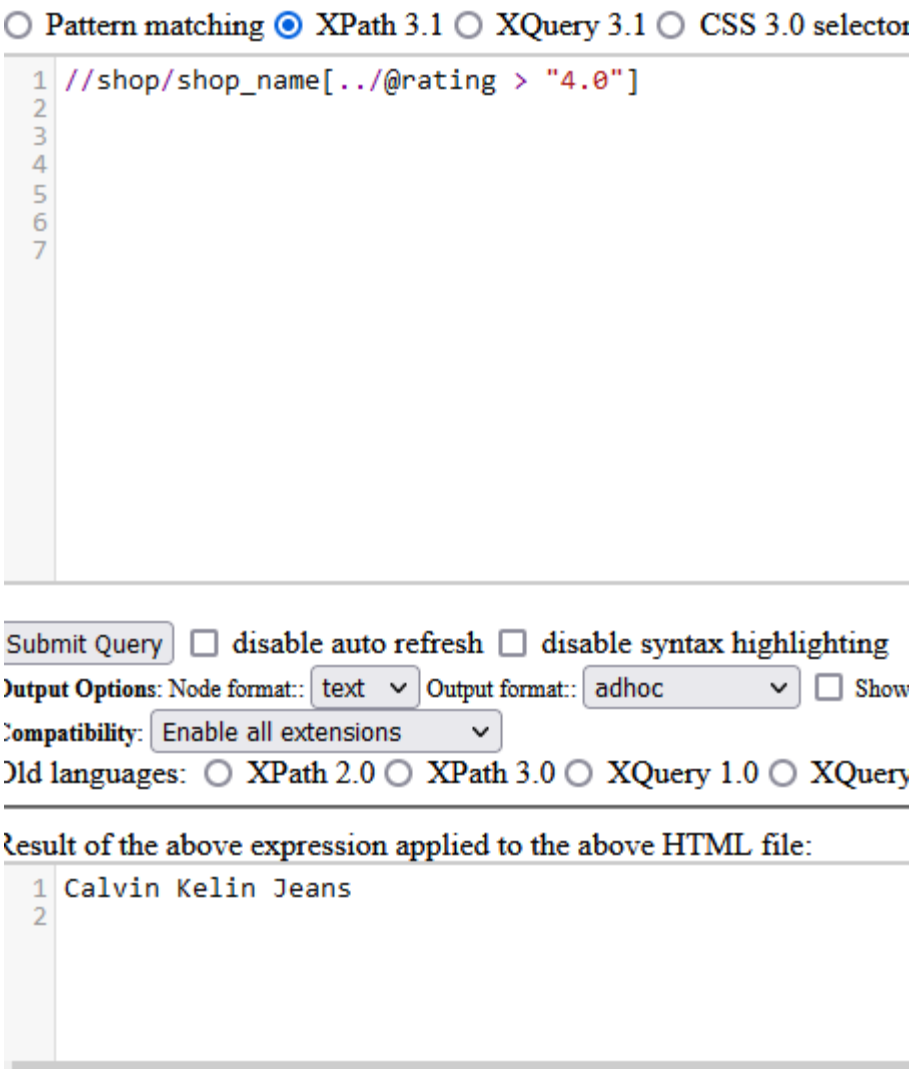
2. Find the floor_no of shop name Calvin Kelin Jeans.
   //shop/floor_no[../shop_name = "Calvin Kelin Jeans"]

   Output:



3. Find the shop name whose rating is greater than 4.
   //shop/shop_name[../@rating > "4.0"]

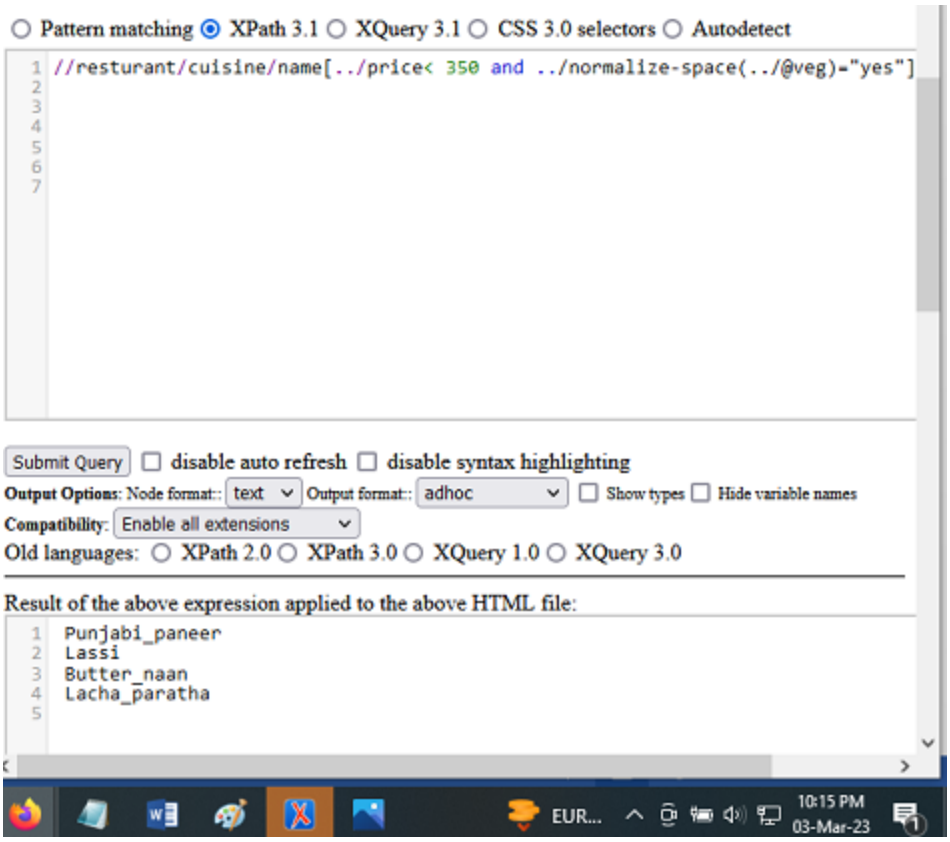
   Output:



4. Name of dishes whose price is less than 350 and whose restaurant type is vegetarian.

//resturant/cuisine/name[../price< 350 and ../normalize-space(../@veg)="yes"]

Output:



5. Find the sum of the price of the product Xiaomi 11T pro 5g and Mi Wired Headset.

sum(//product/product_price[normalize-space(../product_name)="Xiaomi 11T pro 5g" or normalize-space(../product_name)="Mi Wired Headset")

Output:

Pattern matching XPath 3.1 XQuery 3.1 CSS 3.0 selectors Autodetect

```
sum(//product/product_price[normalize-space(../product_name)="Xiaomi 11T pro 5g"
or normalize-space(../product_name)="Mi Wired Headset"])
```

Submit Query disable auto refresh disable syntax highlighting

Output Options: Node format:: text Output format:: adhoc Show types Hide variable names

Compatibility: Enable all extensions

Old languages: XPath 2.0 XPath 3.0 XQuery 1.0 XQuery 3.0

Result of the above expression applied to the above HTML file:

```
37423
```

6. Find all the game names with an age limit of 10.
   for $i in //games
   return $i/game_name[normalize-space(../age_limit) > "10"]

   Output:

Pattern matching XPath 3.1 XQuery 3.1 CSS 3.0 selectors Autodetect

```
for $i in //games
return $i/game_name[normalize-space(../age_limit) > "10"]
```

Submit Query disable auto refresh disable syntax highlighting

Output Options: Node format:: text Output format:: adhoc Show types Hide variable

Compatibility: Enable all extensions

Old languages: XPath 2.0 XPath 3.0 XQuery 1.0 XQuery 3.0

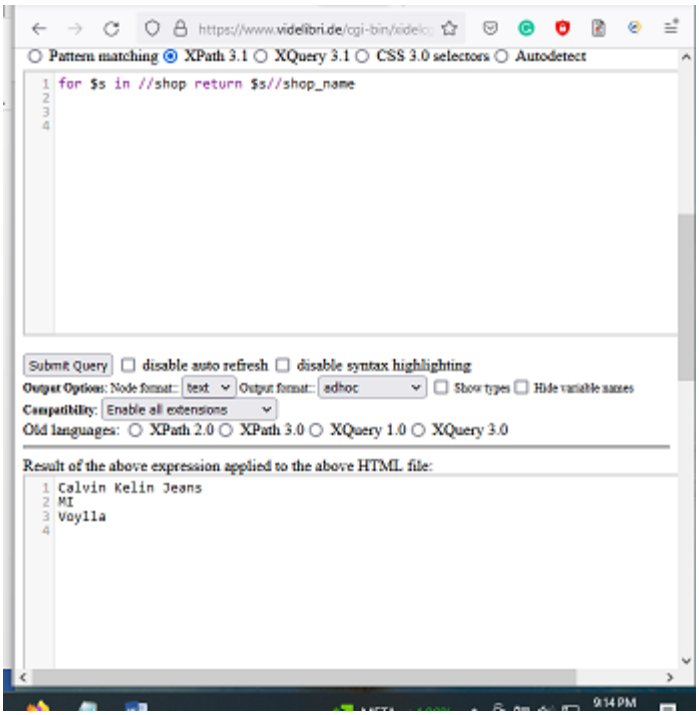
Result of the above expression applied to the above HTML file:

```
 net_cricket
Snooker
```

XQuery

1. return all the shop name
   for $s in //shop return $s//shop_name

   Output:



2. Return the product name whose price is greater than 10000.
   for $x in //shop/product
   where $x/product_price>1000
   return $x/product_name

   Output:

```
for $x in //shop/product
where $x/product_price>1000
return $x/product_name
```

Submit Query ☐ disable auto refresh ☐ disable syntax highlighting

Output Options: Node format:: text Output format:: adhoc ☐ Show types

Compatibility: Enable all extensions

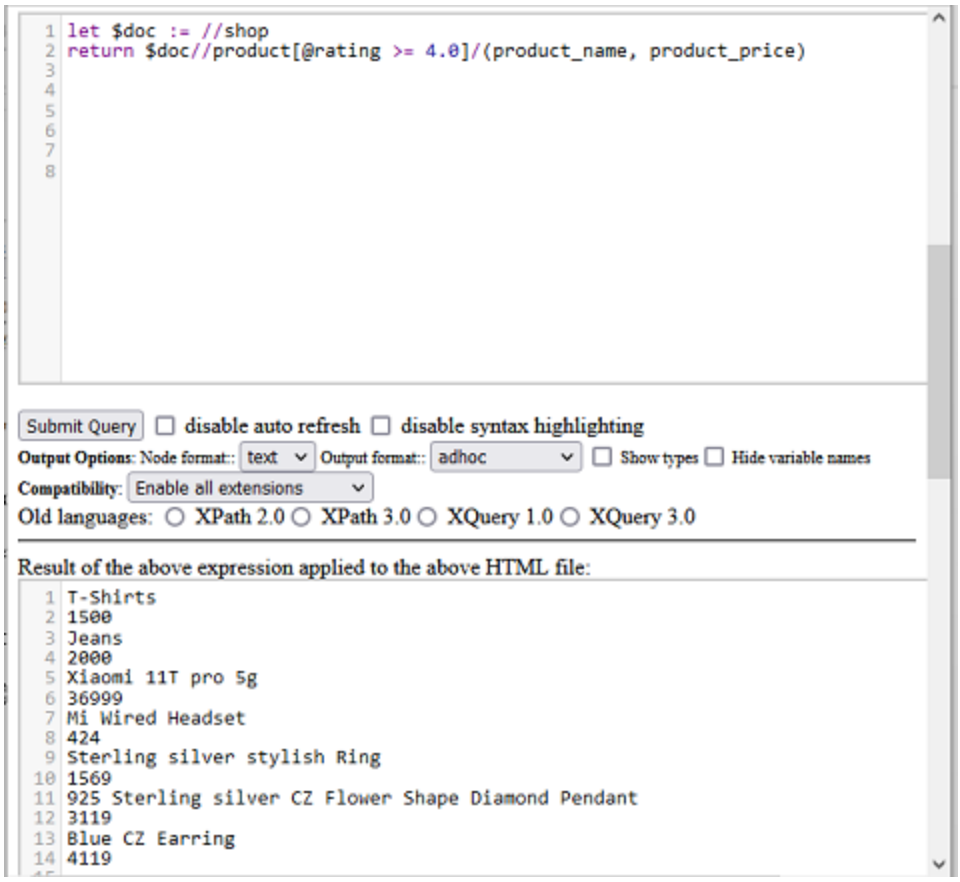
Old languages: ○ XPath 2.0 ○ XPath 3.0 ○ XQuery 1.0 ○ XQuery 3.0

Result of the above expression applied to the above HTML file:

```
T-Shirts
Jeans
Hoodie
Xiaomi 11T pro 5g
Sterling silver stylish Ring
925 Sterling silver CZ Flower Shape Diamond Pendant
Blue CZ Earring
```

3. Get the names and prices of all products with a rating of at least 4.0:
   let $doc := //shop
   return $doc//product[@rating >= 4.0]/(product_name, product_price)

Output:



4. Name of all the restaurants here at keyword can be used to count the iteration.

```
for $x at $i in //resturant
return
<resturant_detail>
        <resturant_name>{$i}. {data($x/res_name)}</resturant_name>
        <resturant_type>{data($x/@veg="yes")}</resturant_type>
        <dish>{data($x/cuisine/name)}</dish>
</resturant_detail>
```

Output:

```
for $x at $i in //resturant
return
<resturant_detail>
    <resturant_name>{$i}. {data($x/res_name)}</resturant_name>
    <resturant_type>{data($x/@veg="yes")}</resturant_type>
    <dish>{data($x/cuisine/name)}</dish>
</resturant_detail>
```

Submit Query ☐ disable auto refresh ☐ disable syntax highlighting

Output Options: Node format:: text Output format:: adhoc ☐ Show types ☐ Hide variable names

Compatibility: Enable all extensions

Old languages: ○ XPath 2.0 ○ XPath 3.0 ○ XQuery 1.0 ○ XQuery 3.0

Result of the above expression applied to the above HTML file:

```
    1. Wow Momo
    false
     veg steamed momos  Burger   chickenwings   cold drink


    2.  Dominos
    false
     margerita pizza   chicken pizza  cheese garlic bread


    3.  punjabi_zaika
    true
     Punjabi_paneer  paneer_achari  Lassi  Butter_naan  Lacha_paratha
```

5. Name of restaurants and type it is veg or non-veg

   for $x in //resturant
   return if ($x/@veg="yes")
   then <veg>{data($x/res_name),data($x/@veg)}</veg>
   else <non_veg>{data($x/res_name),data($x/@veg)}</non_veg>

   Output:

Pattern matching XPath 3.1 XQuery 3.1 CSS 3.0 selectors Autode

```
for $x in //resturant
return if ($x/@veg="yes")
then <veg>{data($x/res_name),data($x/@veg)}</veg>
else <non_veg>{data($x/res_name),data($x/@veg)}</non_veg>
```

Submit Query disable auto refresh disable syntax highlighting

Output Options: Node format:: text Output format:: adhoc Show types Hide

Compatibility: Enable all extensions

Old languages: XPath 2.0 XPath 3.0 XQuery 1.0 XQuery 3.0

Result of the above expression applied to the above HTML file:

```
Wow Momo no
 Dominos no
 punjabi_zaika  yes
```