

**Assignment : 1****XQueries :****1.) Retrieve the usernames and the number of posts for each user:**

```
for $user in //user
return concat("Username: ", $user/@username, ", Number of Posts: ",
count($user/posts/post), "&#xa;")
```

```
1 Username: john_doe, Number of Posts: 3
2
3 Username: , Number of Posts: 0
4
5 Username: , Number of Posts: 0
6
7 Username: jane_smith, Number of Posts: 2
8
9 Username: , Number of Posts: 0
10
11 Username: , Number of Posts: 0
12
13 Username: , Number of Posts: 0
14
15 Username: ashay_shah, Number of Posts: 2
16
17 Username: , Number of Posts: 0
18
19 Username: , Number of Posts: 0
20
21 Username: , Number of Posts: 0
22
23 Username: prince_patel, Number of Posts: 2
24
25 Username: , Number of Posts: 0
26
27 Username: , Number of Posts: 0
28
29 Username: yash_rathore, Number of Posts: 2
30
31 Username: , Number of Posts: 0
32
33 Username: ravi_v, Number of Posts: 2
34
35 Username: , Number of Posts: 0
36
```

```
35 Username: , Number of Posts: 0
36
37 Username: , Number of Posts: 0
38
39 Username: vijay_yadav, Number of Posts: 2
40
41 Username: , Number of Posts: 0
42
43 Username: , Number of Posts: 0
44
45 Username: kuldeep, Number of Posts: 2
46
47 Username: , Number of Posts: 0
48
49 Username: , Number of Posts: 0
50
51 Username: tarun, Number of Posts: 2
52
53 Username: , Number of Posts: 0
54
55 Username: , Number of Posts: 0
56
57 Username: , Number of Posts: 0
58
59 Username: deepak, Number of Posts: 2
60
61 Username: , Number of Posts: 0
62
63 Username: , Number of Posts: 0
64
65
```

The output is a sequence of strings, where each string contains the username and the number of posts for a user, separated by a comma. The `\n` character sequence is used to add a new line after each string.

## 2.) Get the usernames of all users who have at least one friend:

```
for $user in //user[friends/user]
return $user/@username
```

```
1 john_doe
2 jane_smith
3 ashay_shah
4 prince_patel
5 yash_rathore
6 ravi_v
7 vijay_yadav
8 kuldeep
9 tarun
10 deepak
11
```

**3.) Return the first name, last name, and email of the user with id "u002"**

```
let $user := //user[@id = "u002"]
return <user>
  <first_name>{$user/@first_name}</first_name>
  <last_name>{$user/@last_name}</last_name>
  <email>{$user/@email}</email>
</user>
```

```
1 <user>
2   <first_name first_name="Jane"></first_name>
3   <last_name last_name="Smith"></last_name>
4   <email email="janesmith@example.com"></email>
5 </user>
6
```

The output is in xml format.

\$user := //user[@id = "u002"] : Assigns the userid = "u002".

```
return <user>
  <first_name>{$user/@first_name}</first_name>
  <last_name>{$user/@last_name}</last_name>
  <email>{$user/@email}</email>
</user>
```

We are asking the user information in xml format.

**XPath :****1.) Select the id of all users who have the occupation "Therapist"**

/social\_networking\_website/user[profile/@occupation="Therapist"]/@id

```
1 u002
2 u005
3
```

**2.) Select the captions of all images and videos in the third user's posts**

```
/social_networking_website/user[3]/posts/post/*[self::image or self::video]/@caption
```

```
1 At the beach
2 My dog doing tricks
3
```

**3.) Select all the posts posted by user id "u002".**

```
//user[@id="u002"]/posts
```

```
<posts>
  <post id="p004" timestamp="2022-02-14T10:00:00Z">
    <image caption="Happy Valentine's Day!">https://example.com/valentinesday.
  </post>
  <post id="p005" timestamp="2023-03-05T16:45:00Z">
    <video caption="My cat doing tricks">https://example.com/catvideo.mp4</video>
  </post>
</posts>
```