

XPATH

The screenshot displays the XQuery Builder interface. The main editor contains the following XQuery script:

```
<?xml version='1.0' encoding='UTF-8' ?>
<xquery version='3.0' ?>
  let $doc := document('foodDelivery (1).xml')
  let $bill := $doc/bill[mode='online' and status='paid']
  return $bill
end

```

The 'Results' pane on the left shows the output of the query, which is a list of bill records. The 'Project' pane on the right shows the file structure of the project, including 'sample.xml' and 'sample.xpr'.

3. To print the first food item of all restaurant present in the restaurants

The screenshot shows the XML Editor interface with the following components:

- Project:** Lists files like sample.xpr, res, and food_delivery_system.
- Outline:** Shows the structure of the XML document, including food_delivery_system, orders, and order elements.
- Code Editor:** Displays the XML document with the following snippet:

```
<bill bill_number="b7021">
  <amount> 1250 </amount>
  <payment mode="COD" status="pending"/>
</bill>
<order order_id="o504">
  <customer user_id="Abdul101">
    <name> Abdul</name>
    <address>
      <locality> IIT Madras campus </locality>
      <city> Chennai </city>
      <pincode> 600001 </pincode>
    </address>
    <contact_number> 9093459775 </contact_number>
  </customer>
  <food>
    <name> Paneer Chilly </name>
    <quantity> 6 </quantity>
    <price> 40 </price>
  </food>
</order>
```
- Results:** Displays a list of 10 items, including Paneer Chilly, chicken cheese shwama, chicken noodles, maggi, biryani, mutton curry, fish curry, and momo. A tooltip for 'Paneer Chilly' shows 'Press F2 for focus'.
- Footer:** Shows the status bar with 'XPath: successful (0.6s)' and 'U=0000 630 : 1 Modified'.

4. To print the quantity of food items in 5th order

The screenshot shows the XML Editor interface with the following components:

- Project:** Lists files like sample.xpr, res, and food_delivery_system.
- Outline:** Shows the structure of the XML document, including food_delivery_system, orders, and order elements.
- Code Editor:** Displays the XML document with the following snippet:

```
<name> coke 100ml </name>
<quantity> 6 </quantity>
<price> 40 </price>
</food_item>
</food>
<bill bill_number="b7022">
  <amount> 1260 </amount>
  <payment mode="online" status="paid"/>
</bill>
</order>
<order order_id="o505">
  <customer user_id="piyush101">
    <name> Piyush </name>
    <address>
      <locality> Amade Nager </locality>
```
- Results:** Displays a list of 4 items, including coke 100ml, chicken cheese shwama, chicken noodles, maggi, biryani, mutton curry, fish curry, and momo. A tooltip for 'coke 100ml' shows 'Press F2 for focus'.
- Footer:** Shows the status bar with 'XPath: successful (0.6s)' and 'U=000A 196 : 17 Modified'.

Xquery

1.To find the restaurants which are present in city jabalpur

The screenshot shows the XML Editor interface with the following components:

- Project:** sample.xpr
- Outline:** food_delivery_system
 - orders
 - users
 - user "shashank101"
 - user "rohit434"
- Code Editor:** food_delivery_system.xml
 - 39: <?xml version="1.0" encoding="UTF-8"?>
 - 40: <address>
 - 41: <locality> 63 Dukan </locality>
 - 42: <city> jabalpur </city>
 - 43: <pincode> 482001 </pincode>
 - 44: </address>
 - 45: <address>
 - 46: <locality> abajan apartment </locality>
 - 47: <city> surat </city>
 - 48: <pincode> 395500 </pincode>
 - 49: </address>
 - 50: <contact_number> 9123456775 </contact_number>
 - 51: <email> shashank@gmail.com </email>
 - 52: </customer>
 - 53: </delivery>
- XPath/XQuery Builder:** Scope: Current File, Query: /food_delivery_system/restaurants/restaurant/address[contains(., 'jabalpur')]
- Results:**
 - 1. address
 - 2. address

2.To find the delivery boy whose rating is greater >=3

The screenshot shows the XML Editor interface with the following components:

- Project:** sample.xpr
- Outline:** food_delivery_system
 - orders
 - users
 - user "shashank101"
 - user "rohit434"
- Code Editor:** food_delivery_system.xml (same as above)
- XPath/XQuery Builder:** Scope: Current File, Query: /food_delivery_system/orders/order/delivery_delivery_boy/rating>=3]
- Results:**
 - 1. delivery_boy
 - 2. delivery_boy
 - 3. delivery_boy
 - 4. delivery_boy
 - 5. delivery_boy
 - 6. delivery_boy
 - 7. delivery_boy
 - 8. delivery_boy
 - 9. delivery_boy
 - 10. delivery_boy

3. To find the second child of the third order child of orders

The screenshot shows the XML Editor interface with the following components:

- Project:** Lists files like sample.xpr, css, and a debugger.
- Outline:** Shows the XML structure with elements like `food_delivery_system`, `orders`, and `order "0501"`.
- Text View:** Displays the XML content, including `<food item>` and `<bill>` elements.
- XPath/XQuery Builder:** Shows the query: `1 doc("foodDelivery (1).xml") food_delivery_system/orders/2 order[3][2]`.
- Results:** Displays the result of the query, which is the second child of the third order child of orders.

4. To find the last order placed in orders

The screenshot shows the XML Editor interface with the following components:

- Project:** Lists files like sample.xpr, css, and a debugger.
- Outline:** Shows the XML structure with elements like `food_delivery_system`, `orders`, and `order "0501"`.
- Text View:** Displays the XML content, including `<bill>` and `<order>` elements.
- XPath/XQuery Builder:** Shows the query: `1 doc("foodDelivery (1).xml") food_delivery_system/orders/last()`.
- Results:** Displays the result of the query, which is the last order placed in orders.

5. To find the name of food items whose price is less than 100

The screenshot displays the XML Editor interface with the following components:

- Project:** Shows a project named 'sample.xpr' with files 'css', 'debugger', and 'The Main Files support is disabled.'.
- Outline:** Displays the XML structure with elements like 'food_delivery_system', 'orders', 'customer', 'name', 'address', 'locality', 'city', 'pincode', and 'contact_number'.
- XML Document:** Shows the XML content with line numbers 253 to 274. The XML structure includes a delivery boy, a restaurant, and food items with their names, quantities, and prices.
- XPath/XQuery Builder:** Shows the query:

```
1 for $x in doc("foodDelivery (1).xml")/food_delivery_system/orders/order/food_food_item
2 where $x/price < 100
3 return $x/name
```
- Results:** Displays the results of the query, showing the names of food items whose price is less than 100: 1. name, 2. name, 3. name, 4. name, 5. name, 6. name, 7. name, 8. name, 9. name.

The bottom status bar indicates the file path 'D:\Tanu\TAO\Assignmet2\foodDelivery (1).xml' and the XPath query is successful.