# Policy Bazaar Home Page Locator Design & Testing

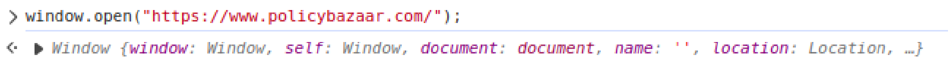
**Task 1:**

- Visit Policy Bazaar Home Page:
- Write a script to navigate to the Policy Bazaar home page (https://www.policybazaar.com/).

**Solution:**

This Problem can be handled by many ways:

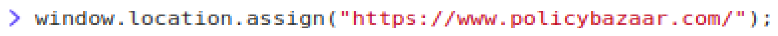
1. By "window.open()" method we can easily navigate to the website in the new tab.



```
> window.open("https://www.policybazaar.com/");
<· ▶ Window {window: Window, self: Window, document: document, name: '', location: Location, …}
```

- **window:** 'window' refer here to the global browser window object.
- **Open():** 'open()' is a method of the window object that is used to open a new browser window or tab.
- We can passed the argument in the open() method and navigate to the browser through the console.

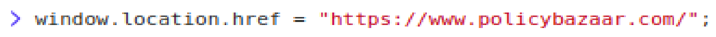
2. By "window.location.assign()" method we can easily naviagate to the website in the current browsing context.



```
> window.location.assign("https://www.policybazaar.com/");
```

- **window.location:** 'window.location' is the property of the window object that is represent the current URL of the browser.
- **assign(): This is the method of** 'window.location' object, it takes the URL as the argument and loads the URL in the current browsing window.

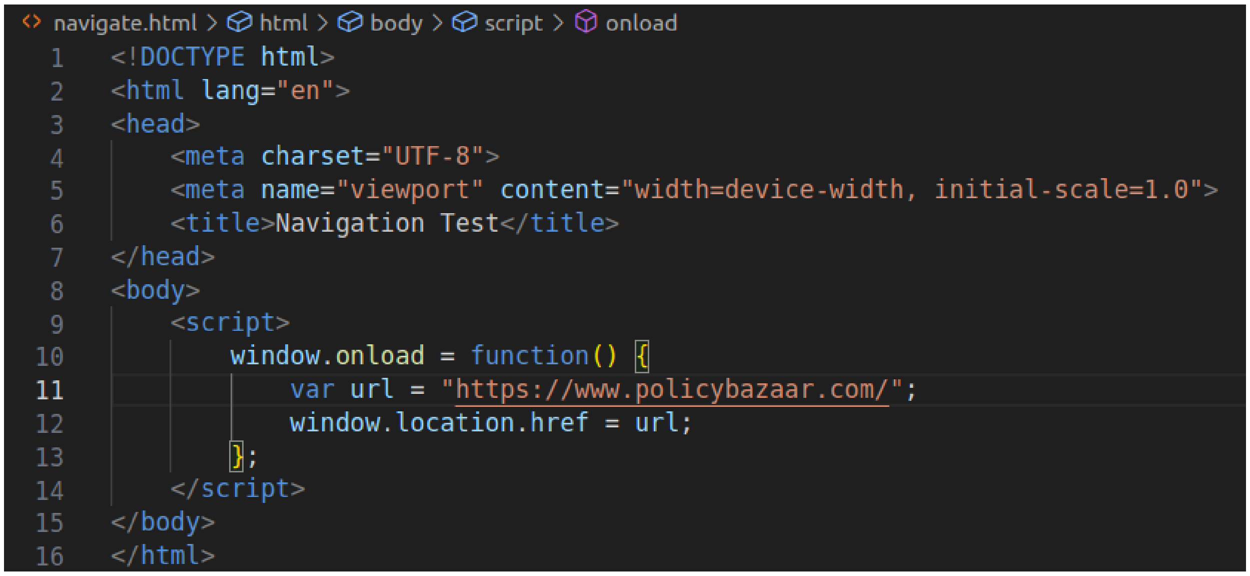
3. By "window.location.href = "URL"  is the property of the 'window.location' object that represent the current URL in the current page.



```
> window.location.href = "https://www.policybazaar.com/";
```

- 
- **window.location:** 'window.location' is the property of the window object that is represent the current URL of the browser.
- **.href:** It is a property of the window.location object that represents the complete URL of the current page

4. By using IDE (vs code):



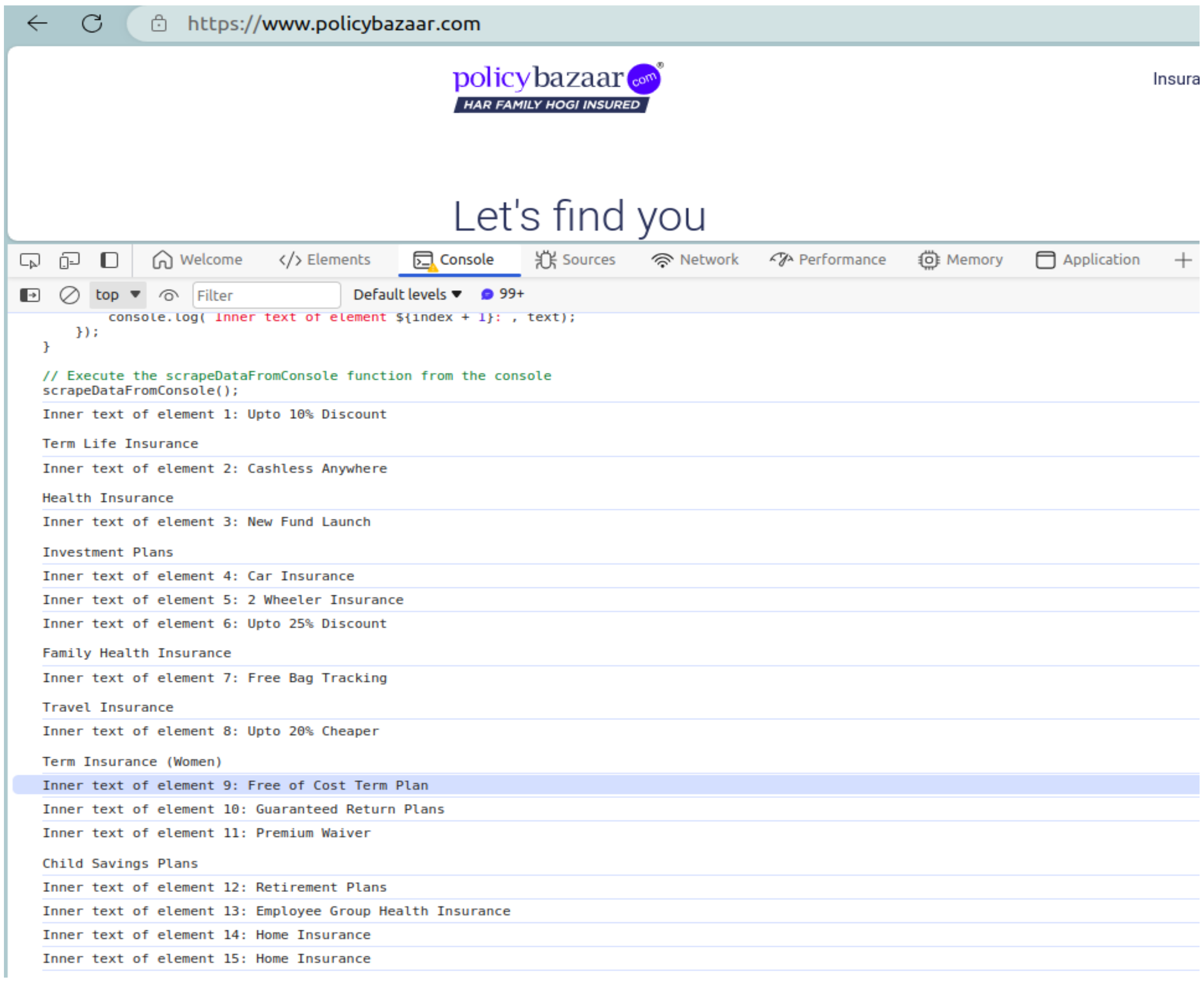
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Navigation Test</title>
</head>
<body>
    <script>
        window.onload = function() {
            var url = "https://www.policybazaar.com/";
            window.location.href = url;
        };
    </script>
</body>
</html>
```

**Task 2:**

- Design locators to extract the inner text of each insurance product listed on the home page.

```
function scrapeDataFromConsole() {
  // Function to navigate to a specific URL
  function navigateTo(url) {
    window.location.href = url;
  }


  // Function to extract inner text from elements matching a selector
  function extractInnerText(selector) {
    const elements = document.querySelectorAll(selector);
    const innerTextArray = [];
    elements.forEach(function(element, index) {
      innerTextArray.push(element.innerText.trim());
    });
```

```
        return innerTextArray;
    }

    // Navigating to the website
    navigateTo('https://policybazaar.com');
    const innerTextArray = extractInnerText('.prd-icon.add.shadowHandler.short');

    innerTextArray.forEach(function(text, index) {
        console.log(`Inner text of element ${index + 1}:`, text);
    });
}
scrapeDataFromConsole();
```



```
        console.log(`Inner text of element ${index + 1}: `, text);
    });
}

// Execute the scrapeDataFromConsole function from the console
scrapeDataFromConsole();
Inner text of element 1: Upto 10% Discount

Term Life Insurance
Inner text of element 2: Cashless Anywhere

Health Insurance
Inner text of element 3: New Fund Launch

Investment Plans
Inner text of element 4: Car Insurance
Inner text of element 5: 2 Wheeler Insurance
Inner text of element 6: Upto 25% Discount

Family Health Insurance
Inner text of element 7: Free Bag Tracking

Travel Insurance
Inner text of element 8: Upto 20% Cheaper

Term Insurance (Women)
Inner text of element 9: Free of Cost Term Plan
Inner text of element 10: Guaranteed Return Plans
Inner text of element 11: Premium Waiver

Child Savings Plans
Inner text of element 12: Retirement Plans
Inner text of element 13: Employee Group Health Insurance
Inner text of element 14: Home Insurance
Inner text of element 15: Home Insurance
```

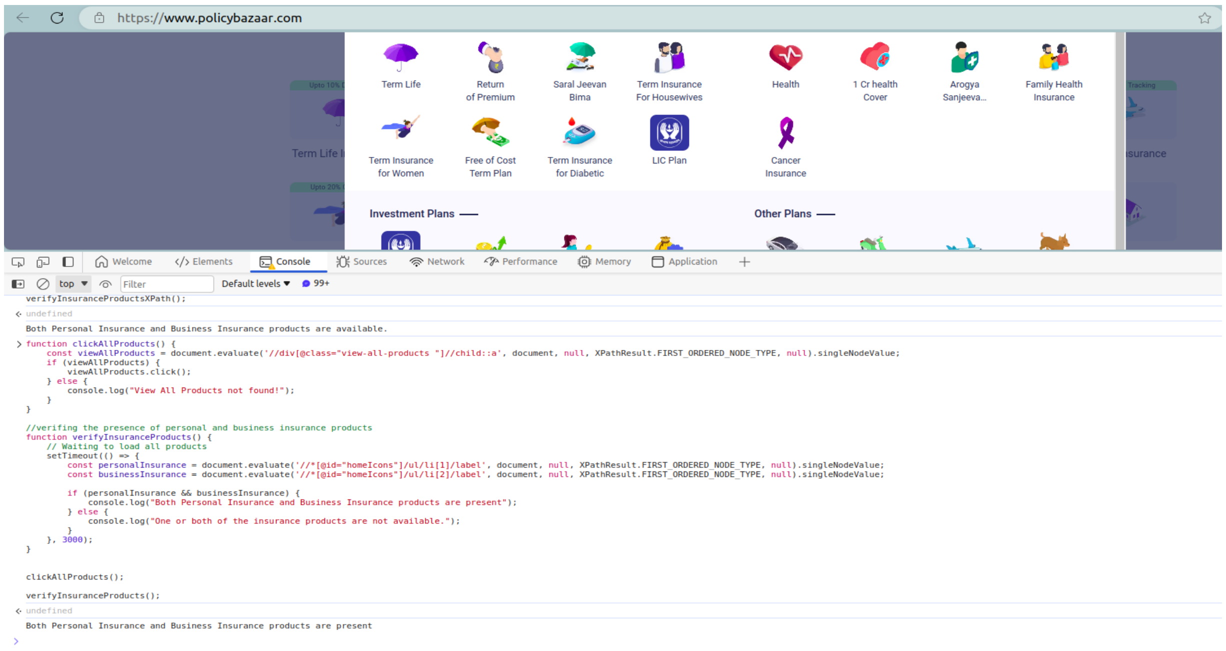
**Task 03:**

- Click on "View All Products"
- Verify there are two types of insurance products. ** Personal Insurance ** Business Insurance

**Solution:**

```
function clickAllProducts() {
    const viewAllProducts = document.evaluate('//div[@class="view-all-products  "]//child::a', document, null, XPathResult.FIRST_ORDERED_NODE_TYPE, null).singleNodeValue;
    if (viewAllProducts) {
        viewAllProducts.click();
    } else {
        console.log("View All Products not found!");
    }
}
//verifing the presence of personal and business insurance products
function verifyInsuranceProducts() {
    // Waiting to load all products
    setTimeout(() => {
        const personalInsurance = document.evaluate('//*[@id="homeIcons"]/ul/li[1]/label', document, null, XPathResult.FIRST_ORDERED_NODE_TYPE, null).singleNodeValue;
        const businessInsurance = document.evaluate('//*[@id="homeIcons"]/ul/li[2]/label', document, null, XPathResult.FIRST_ORDERED_NODE_TYPE, null).singleNodeValue;

        if (personalInsurance && businessInsurance) {
            console.log("Both Personal Insurance and Business Insurance products are present");
        } else {
            console.log("One or both of the insurance products are not available.");
        }
    }, 3000);
}
clickAllProducts();
verifyInsuranceProducts();
```

**Screenshot:**



```
verifyInsuranceProductsXPath();
<· undefined
Both Personal Insurance and Business Insurance products are available.
> function clickAllProducts() {
      const viewAllProducts = document.evaluate('//div[@class="view-all-products "]//child::a', document, null, XPathResult.FIRST_ORDERED_NODE_TYPE, null).singleNodeValue;
      if (viewAllProducts) {
          viewAllProducts.click();
      } else {
          console.log("View All Products not found!");
      }
  }

  //verifing the presence of personal and business insurance products
  function verifyInsuranceProducts() {
      // Waiting to load all products
      setTimeout(() => {
          const personalInsurance = document.evaluate('//*[@id="homeIcons"]/ul/li[1]/label', document, null, XPathResult.FIRST_ORDERED_NODE_TYPE, null).singleNodeValue;
          const businessInsurance = document.evaluate('//*[@id="homeIcons"]/ul/li[2]/label', document, null, XPathResult.FIRST_ORDERED_NODE_TYPE, null).singleNodeValue;

          if (personalInsurance && businessInsurance) {
              console.log("Both Personal Insurance and Business Insurance products are present");
          } else {
              console.log("One or both of the insurance products are not available.");
          }
      }, 3000);
  }


  clickAllProducts();

  verifyInsuranceProducts();
<· undefined
Both Personal Insurance and Business Insurance products are present
>
```

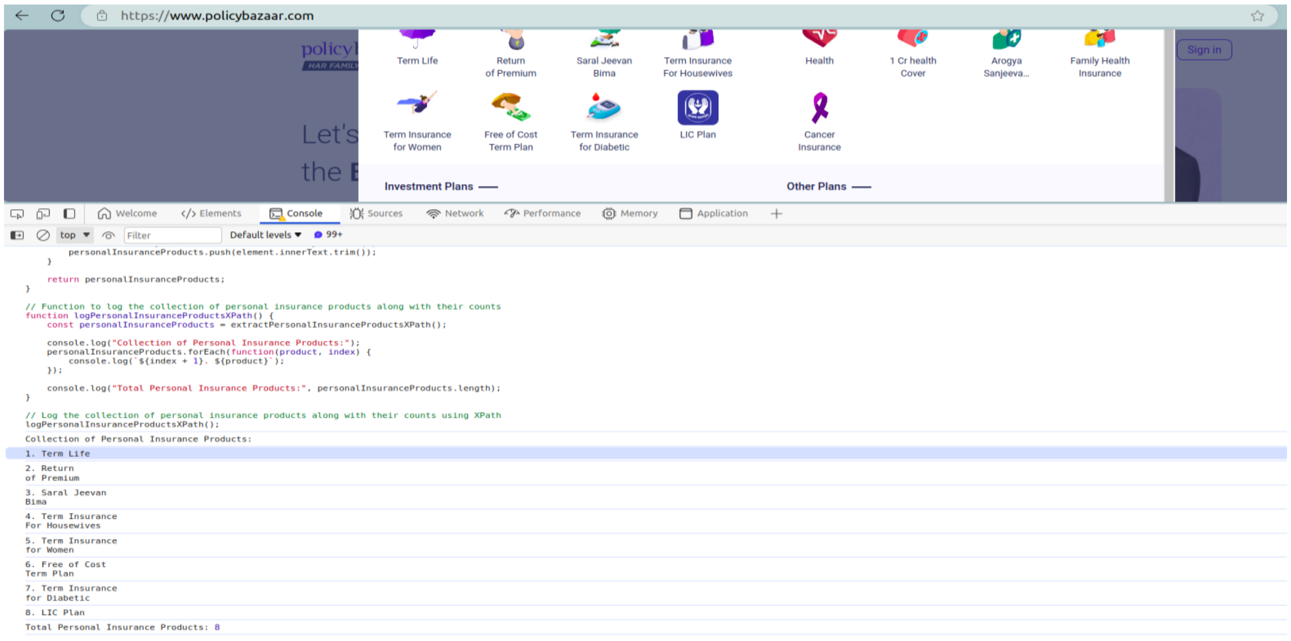
**Task 4:**

- Design locators to extract inner text information for each personal insurance product.
- Log the collection of personal insurance products along with their counts.

**Solution:** As I examined, there are **four sections** in the Personal Insurance section & we can extract each personal Insurance Product and get the inner text by following the modular approach in all sections and create a function for each section of it.

**1. For Term Insurance ( Personal Insurance)**

```
function extractPersonalInsuranceProducts() {
    const personalInsuranceProducts = [];
// Xpath locator for this operation
const personalInsuranceElements = document.evaluate('//div[@id="tab-content1"]//child::div[1]//child::div[1]//child::ul[1]//child::a[1]', document, null, XPathResult.ORDERED_NODE_SNAPSHOT_TYPE, null);


    for (let i = 0; i < personalInsuranceElements.snapshotLength; i++) {
        const element = personalInsuranceElements.snapshotItem(i);
        personalInsuranceProducts.push(element.innerText.trim());
    }
    return personalInsuranceProducts;
}


// Function to log the collection of personal insurance products along with their counts
function logPersonalInsuranceProductsXPath() {
    const personalInsuranceProducts = extractPersonalInsuranceProductsXPath();


    console.log("Collection of Personal Insurance Products:");
    personalInsuranceProducts.forEach(function(product, index) {
        console.log(`${index + 1}. ${product}`);
    });
    console.log("Total Personal Insurance Products:", personalInsuranceProducts.length);
}
logPersonalInsuranceProducts();
```



```
        personalInsuranceProducts.push(element.innerText.trim());
    }

    return personalInsuranceProducts;
}

// Function to log the collection of personal insurance products along with their counts
function logPersonalInsuranceProductsXPath() {
    const personalInsuranceProducts = extractPersonalInsuranceProductsXPath();

    console.log("Collection of Personal Insurance Products:");
    personalInsuranceProducts.forEach(function(product, index) {
        console.log(`${index + 1}. ${product}`);
    });

    console.log("Total Personal Insurance Products:", personalInsuranceProducts.length);
}

// Log the collection of personal insurance products along with their counts using XPath
logPersonalInsuranceProductsXPath();
Collection of Personal Insurance Products:
1. Term Life
2. Return
of Premium
3. Saral Jeevan
Bima
4. Term Insurance
For Housewives
5. Term Insurance
for Women
6. Free of Cost
Term Plan
7. Term Insurance
for Diabetic
8. LIC Plan
Total Personal Insurance Products: 8
```

**Explanation:**

- **xpathresult.ordered_node_snapshot_type:** This gets a list of nodes from an XPath query. The nodes are in the same order as they appear in the document.

- **snapshotItem(i):** This is a function that lets we access a specific node in the list by its index i.

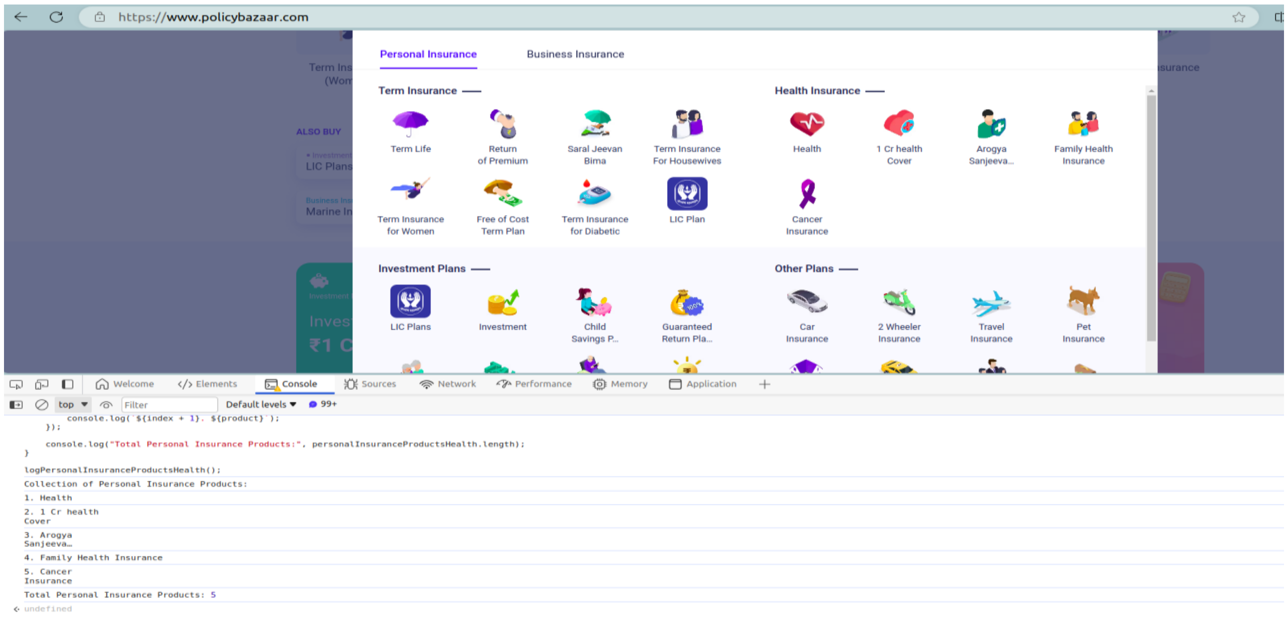
- **snapshotLength:** This tells us how many nodes are in the list.

So, if we run an XPath query and use **ordered_node_snapshot_type,** we get a list of nodes. We can find out how many nodes are in the list using **snapshotLength,** and access each node using **snapshotItem(i).** If the document changes, our list stays the same because it's a "snapshot".

**2. For Health(Personal Insuance)**

```
function extractPersonalInsuranceProductsHealth() {
  const personalInsuranceProductsHealth = [];
  const personalInsuranceElementsHealth = document.evaluate('//*[@id="tab-content1"]//child::div[1]//child::div[2]/child::ul[1]//child::li', document, null, XPathResult.ORDERED_NODE_SNAPSHOT_TYPE, null);

  for (let i = 0; i < personalInsuranceElementsHealth.snapshotLength; i++) {
    const element = personalInsuranceElementsHealth.snapshotItem(i);
    personalInsuranceProductsHealth.push(element.innerText.trim());
  }

  return personalInsuranceProductsHealth;
}

// Function to log the collection of personal insurance products
```

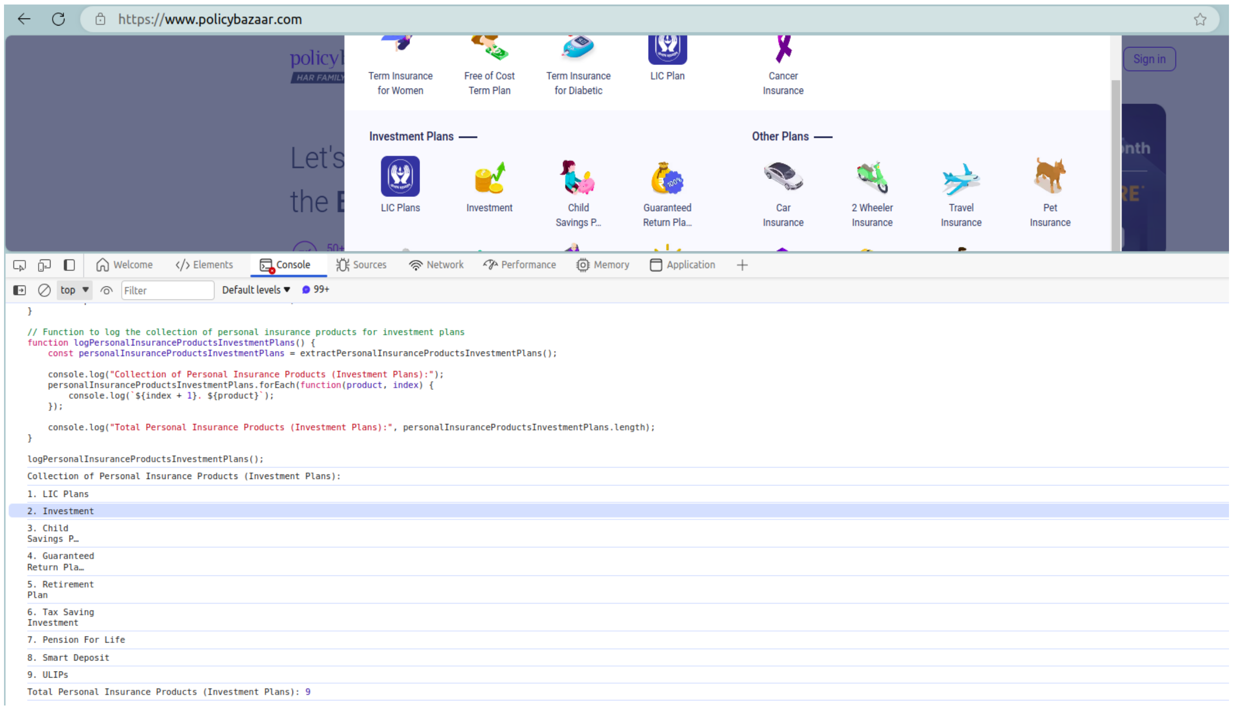
```
function logPersonalInsuranceProductsHealth() {
    const personalInsuranceProductsHealth = extractPersonalInsuranceProducts();

    console.log("Collection of Personal Insurance Products:");
    personalInsuranceProductsHealth.forEach(function(product, index) {
        console.log(`${index + 1}. ${product}`);
    });

    console.log("Total Personal Insurance Products:", personalInsuranceProductsHealth.length);
}

logPersonalInsuranceProductsHealth();
```



```
        console.log(`${index + 1}. ${product}`);
    });

    console.log("Total Personal Insurance Products:", personalInsuranceProductsHealth.length);
}

logPersonalInsuranceProductsHealth();
Collection of Personal Insurance Products:
1. Health
2. 1 Cr health
Cover
3. Arogya
Sanjeeva…
4. Family Health Insurance
5. Cancer
Insurance
Total Personal Insurance Products: 5
← undefined
```

## 3. For Investment Plans (Personal Insurance)

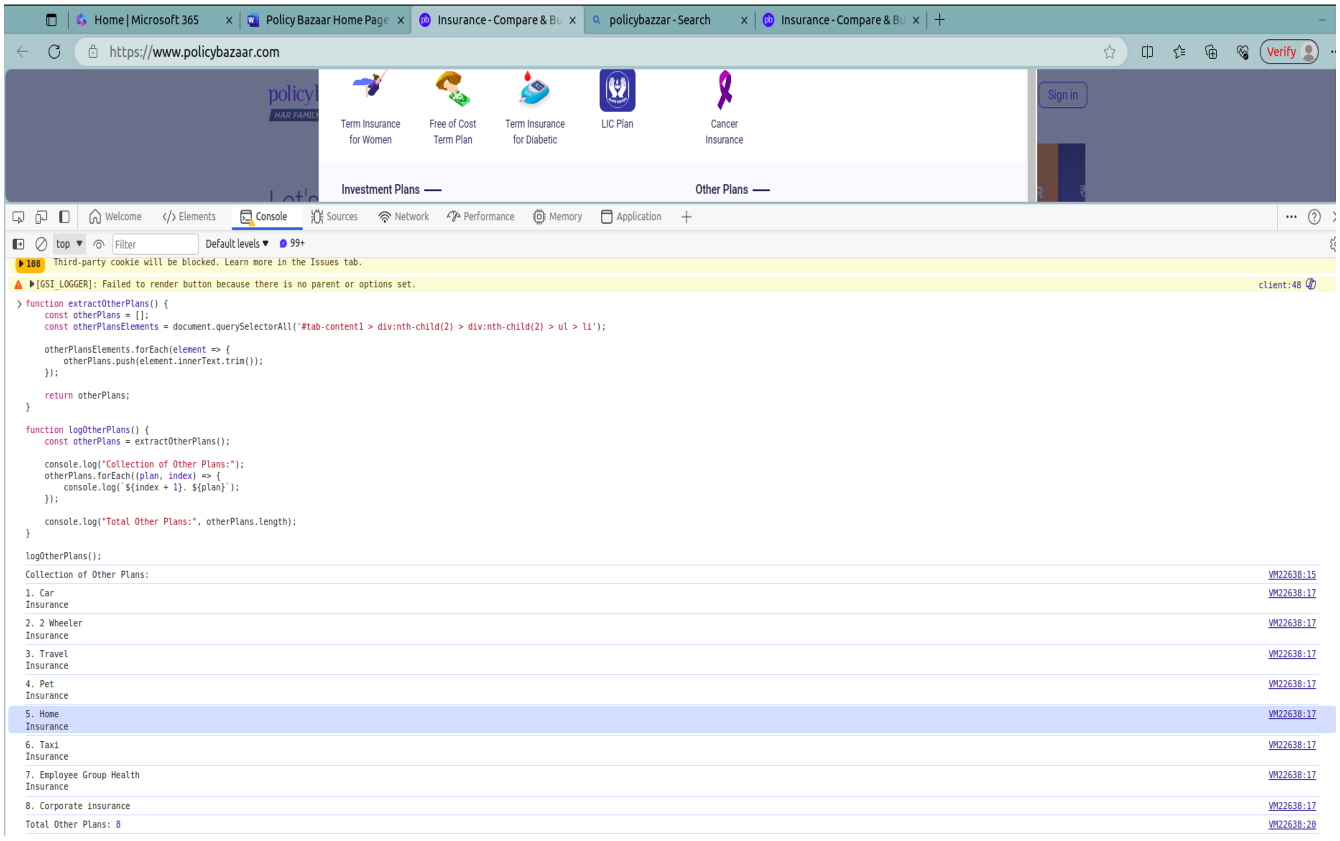
```
function extractPersonalInsuranceProductsInvestmentPlans() {
    const personalInsuranceProductsInvestmentPlans = [];
    const personalInsuranceElementsInvestmentPlans = document.evaluate("//div[@class = 'icon-block-row bg-blue']//child::div[1]//child::li ", document, null, XPathResult.ORDERED_NODE_SNAPSHOT_TYPE, null);
    for (let i = 0; i < personalInsuranceElementsInvestmentPlans.snapshotLength; i++) {
        const element = personalInsuranceElementsInvestmentPlans.snapshotItem(i);
        personalInsuranceProductsInvestmentPlans.push(element.innerText.trim());
    }

    return personalInsuranceProductsInvestmentPlans;
}

// Function to log the collection of personal insurance products for investment plans
function logPersonalInsuranceProductsInvestmentPlans() {
const personalInsuranceProductsInvestmentPlans =
extractPersonalInsuranceProductsInvestmentPlans();

    console.log("Collection of Personal Insurance Products (Investment Plans):");
    personalInsuranceProductsInvestmentPlans.forEach(function(product, index) {
        console.log(`${index + 1}. ${product}`);
    });

    console.log("Total Personal Insurance Products (Investment Plans):",
personalInsuranceProductsInvestmentPlans.length);
}

logPersonalInsuranceProductsInvestmentPlans();
```



```
}

// Function to log the collection of personal insurance products for investment plans
function logPersonalInsuranceProductsInvestmentPlans() {
    const personalInsuranceProductsInvestmentPlans = extractPersonalInsuranceProductsInvestmentPlans();

    console.log("Collection of Personal Insurance Products (Investment Plans):");
    personalInsuranceProductsInvestmentPlans.forEach(function(product, index) {
        console.log(`${index + 1}. ${product}`);
    });

    console.log("Total Personal Insurance Products (Investment Plans):", personalInsuranceProductsInvestmentPlans.length);
}

logPersonalInsuranceProductsInvestmentPlans();
Collection of Personal Insurance Products (Investment Plans):
1. LIC Plans
2. Investment
3. Child
Savings P…
4. Guaranteed
Return Pla…
5. Retirement
Plan
6. Tax Saving
Investment
7. Pension For Life
8. Smart Deposit
9. ULIPs
Total Personal Insurance Products (Investment Plans): 9
```

## 4. For Other Plans (Personal Insurance)

```
function extractOtherPlans() {
    const otherPlans = [];
    const otherPlansElements = document.querySelectorAll('#tab-content1 > div:nth-child(2) > div:nth-child(2) > ul > li');

    otherPlansElements.forEach(element => {
        otherPlans.push(element.innerText.trim());
    });

    return otherPlans;
}

function logOtherPlans() {
    const otherPlans = extractOtherPlans();
    console.log("Collection of Other Plans:");
    otherPlans.forEach((plan, index) => {
        console.log(`${index + 1}. ${plan}`);
    });

    console.log("Total Other Plans:", otherPlans.length);
}
logOtherPlans();
```



```
108 Third-party cookie will be blocked. Learn more in the Issues tab.
[GSI_LOGGER]: Failed to render button because there is no parent or options set.        client:48
> function extractOtherPlans() {
      const otherPlans = [];
      const otherPlansElements = document.querySelectorAll('#tab-content1 > div:nth-child(2) > div:nth-child(2) > ul > li');

      otherPlansElements.forEach(element => {
          otherPlans.push(element.innerText.trim());
      });

      return otherPlans;
  }

  function logOtherPlans() {
      const otherPlans = extractOtherPlans();

      console.log("Collection of Other Plans:");
      otherPlans.forEach((plan, index) => {
          console.log(`${index + 1}. ${plan}`);
      });

      console.log("Total Other Plans:", otherPlans.length);
  }

  logOtherPlans();
  Collection of Other Plans:                    VM22638:15
  1. Car
  Insurance                                     VM22638:17
  2. 2 Wheeler
  Insurance                                     VM22638:17
  3. Travel
  Insurance                                     VM22638:17
  4. Pet
  Insurance                                     VM22638:17
  5. Home
  Insurance                                     VM22638:17
  6. Taxi
  Insurance                                     VM22638:17
  7. Employee Group Health
  Insurance                                     VM22638:17
  8. Corporate insurance                        VM22638:17
  Total Other Plans: 8                          VM22638:20
```

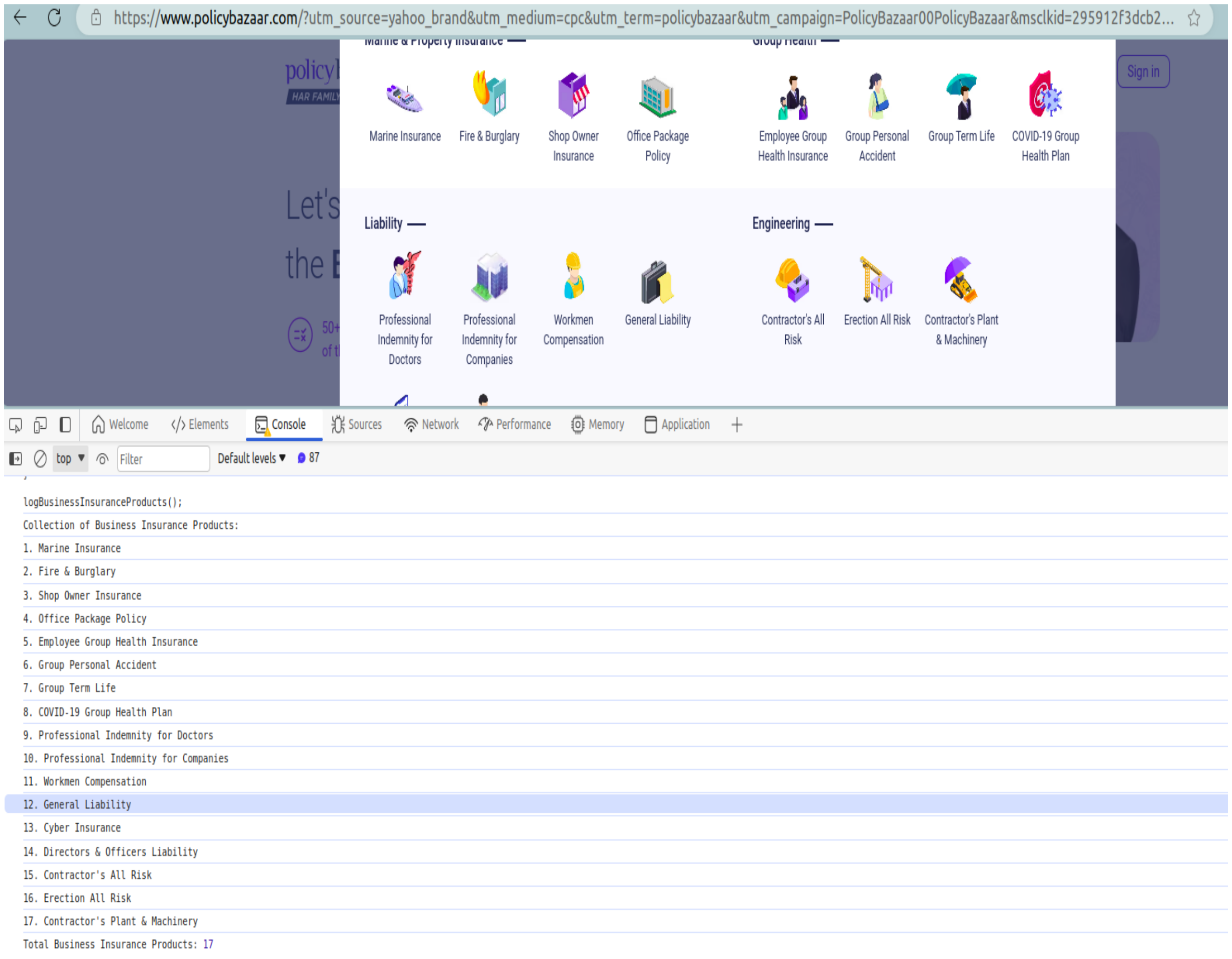
**Task 5:**

- Design locators to extract inner text information for each business insurance product.
- Log the collection of business insurance products along with their counts.

As I examined, we can extract inner parts by two ways first one is extract all products under this section in one time execution & other is by each function.

**First Approach :**

```
// Function to extract business insurance products using the provided XPath expression

function extractBusinessInsurance() {

    // Array to store business insurance products

    const businessInsuranceProducts = [];

    // Evaluate the XPath expression to select relevant elements

    const businessInsuranceElements = document.evaluate("//li[@class='tab']//div[@id='tab-
content2']//a",document,null,XPathResult.ORDERED_NODE_SNAPSHOT_TYPE,  null);

    // Iterate through selected elements and push their text content into the array

    for (let i = 0; i < businessInsuranceElements.snapshotLength; i++) {

        const element = businessInsuranceElements.snapshotItem(i);

        businessInsuranceProducts.push(element.innerText.trim());

    }


    // Return the array of business insurance products

    return businessInsuranceProducts;

}


// Function to log the collection of business insurance products

function logBusinessInsuranceProducts() {

    // Extract business insurance products

    const businessInsuranceProducts = extractBusinessInsurance();


    // Log the collection of business insurance products

    console.log("Collection of Business Insurance Products:");
    businessInsuranceProducts.forEach((product, index) => {
        console.log(`${index + 1}. ${product}`);
    });

    // Log the total number of business insurance products
    console.log("Total Business Insurance Products:", businessInsuranceProducts.length);
}

// Call the function to log business insurance products
logBusinessInsuranceProducts();
```

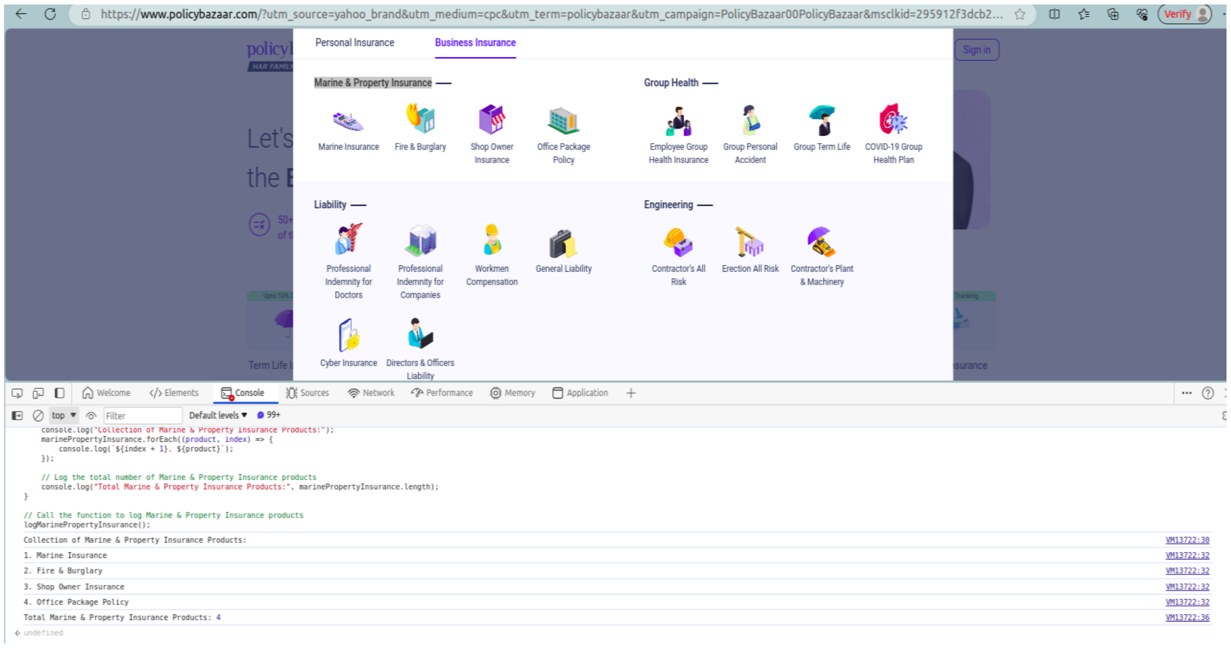

https://www.policybazaar.com/?utm_source=yahoo_brand&utm_medium=cpc&utm_term=policybazaar&utm_campaign=PolicyBazaar00PolicyBazaar&msclkid=295912f3dcb2...
Marine & Property Insurance
Group Health
Sign in
Marine Insurance
Fire & Burglary
Shop Owner Insurance
Office Package Policy
Employee Group Health Insurance
Group Personal Accident
Group Term Life
COVID-19 Group Health Plan
Liability
Engineering
Professional Indemnity for Doctors
Professional Indemnity for Companies
Workmen Compensation
General Liability
Contractor's All Risk
Erection All Risk
Contractor's Plant & Machinery
Welcome
Elements
Console
Sources
Network
Performance
Memory
Application
top
Filter
Default levels
87

```
logBusinessInsuranceProducts();
Collection of Business Insurance Products:
1. Marine Insurance
2. Fire & Burglary
3. Shop Owner Insurance
4. Office Package Policy
5. Employee Group Health Insurance
6. Group Personal Accident
7. Group Term Life
8. COVID-19 Group Health Plan
9. Professional Indemnity for Doctors
10. Professional Indemnity for Companies
11. Workmen Compensation
12. General Liability
13. Cyber Insurance
14. Directors & Officers Liability
15. Contractor's All Risk
16. Erection All Risk
17. Contractor's Plant & Machinery
Total Business Insurance Products: 17
```

**Second Approach: By Each Section**

**1. Marine & Property Insurance:**

```
// Function to extract Marine & Property Insurance products using the provided XPath expression
function extractMarinePropertyInsurance() {
    // Array to store Marine & Property Insurance products
    const marinePropertyInsurance = [];
    // Evaluate the XPath expression to select relevant elements
    const insuranceElements = document.evaluate(
        '//div[@id="tab-content2"]//child::div[1]/child::div[1]/child::ul[1]/li',
        document,
        null,
        XPathResult.ORDERED_NODE_SNAPSHOT_TYPE,
        null
    );

    // Iterate through selected elements and push their text content into the array
    for (let i = 0; i < insuranceElements.snapshotLength; i++) {
        const element = insuranceElements.snapshotItem(i);
        marinePropertyInsurance.push(element.innerText.trim());
    }

    // Return the array of Marine & Property Insurance products
    return marinePropertyInsurance;
}

// Function to log the collection of Marine & Property Insurance products
function logMarinePropertyInsurance() {
    // Extract Marine & Property Insurance products
    const marinePropertyInsurance = extractMarinePropertyInsurance();

    // Log the collection of Marine & Property Insurance products
    console.log("Collection of Marine & Property Insurance Products:");
    marinePropertyInsurance.forEach((product, index) => {
        console.log(`${index + 1}. ${product}`);
    });

    // Log the total number of Marine & Property Insurance products
   console.log("Total Marine & Property Insurance Products:", marinePropertyInsurance.length);
}

// Call the function to log Marine & Property Insurance products
logMarinePropertyInsurance();
```



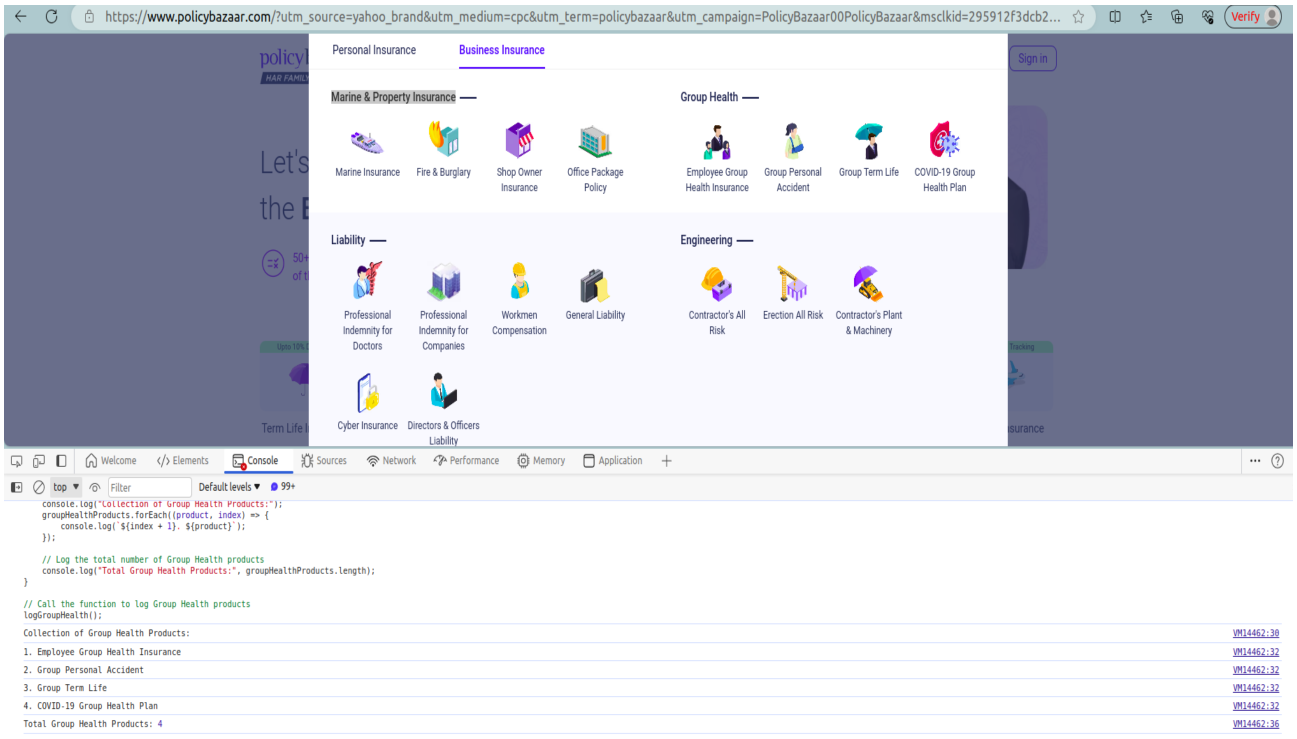
https://www.policybazaar.com/?utm_source=yahoo_brand&utm_medium=cpc&utm_term=policybazaar&utm_campaign=PolicyBazaar00PolicyBazaar&msclkid=295912f3dcb2...
Personal Insurance
Business Insurance
Marine & Property Insurance
Marine Insurance
Fire & Burglary
Shop Owner Insurance
Office Package Policy
Group Health
Employee Group Health Insurance
Group Personal Accident
Group Term Life
COVID-19 Group Health Plan
Liability
Professional Indemnity for Doctors
Professional Indemnity for Companies
Workmen Compensation
General Liability
Cyber Insurance
Directors & Officers Liability
Engineering
Contractor's All Risk
Erection All Risk
Contractor's Plant & Machinery
Welcome
Elements
Console
Sources
Network
Performance
Memory
Application

```
    console.log("Collection of Marine & Property Insurance Products:");
    marinePropertyInsurance.forEach((product, index) => {
        console.log(`${index + 1}. ${product}`);
    });

    // Log the total number of Marine & Property Insurance products
    console.log("Total Marine & Property Insurance Products:", marinePropertyInsurance.length);
}

// Call the function to log Marine & Property Insurance products
logMarinePropertyInsurance();
Collection of Marine & Property Insurance Products:                VM13722:30
1. Marine Insurance                                                VM13722:32
2. Fire & Burglary                                                 VM13722:32
3. Shop Owner Insurance                                            VM13722:32
4. Office Package Policy                                           VM13722:32
Total Marine & Property Insurance Products: 4                      VM13722:36
← undefined
```

## 2. Group Health

// Function to extract Group Health products using the provided XPath expression

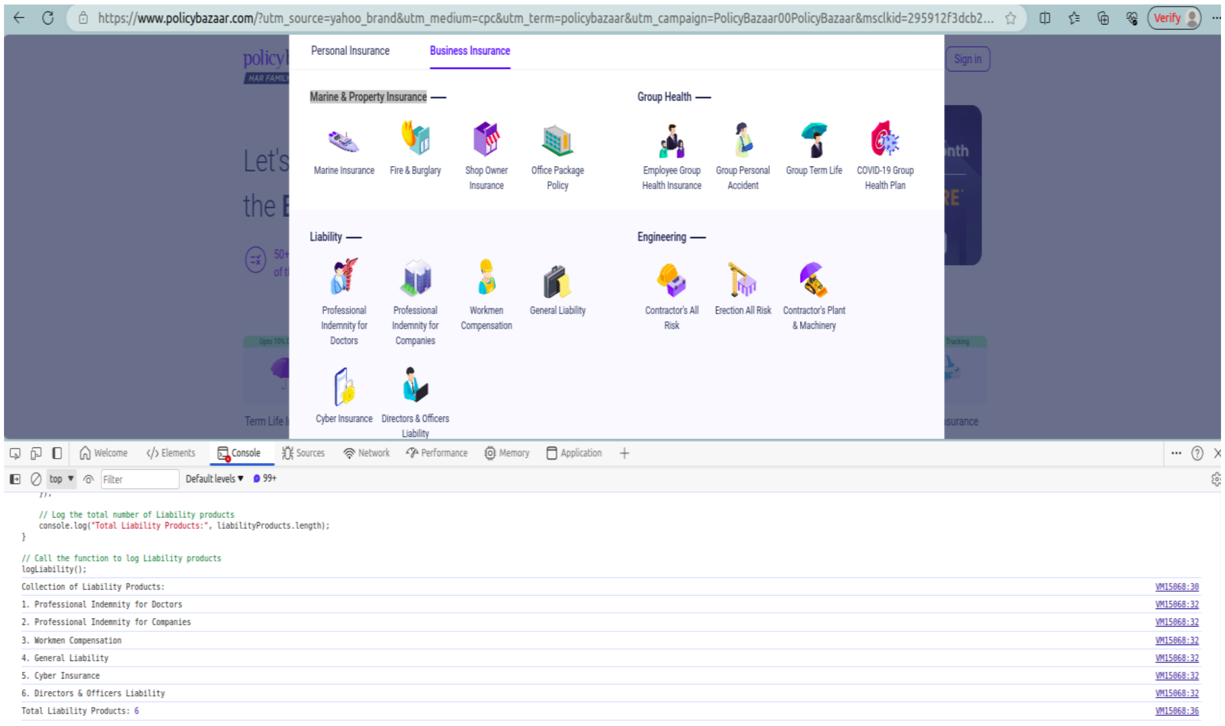
```
function extractGroupHealth() {
    // Array to store Group Health products
    const groupHealthProducts = [];
    // Evaluate the XPath expression to select relevant elements
    const groupHealthElements = document.evaluate(
        '//div[@id="tab-content2"]//child::div[1]/child::div[2]/child::ul[1]/li',
        document,
        null,
        XPathResult.ORDERED_NODE_SNAPSHOT_TYPE,
        null
    );

    // Iterate through selected elements and push their text content into the array
    for (let i = 0; i < groupHealthElements.snapshotLength; i++) {
        const element = groupHealthElements.snapshotItem(i);
        groupHealthProducts.push(element.innerText.trim());
    }

    // Return the array of Group Health products
    return groupHealthProducts;
}

// Function to log the collection of Group Health products
function logGroupHealth() {
    // Extract Group Health products
    const groupHealthProducts = extractGroupHealth();

    // Log the collection of Group Health products
    console.log("Collection of Group Health Products:");
    groupHealthProducts.forEach((product, index) => {
        console.log(`${index + 1}. ${product}`);
    });

    // Log the total number of Group Health products
    console.log("Total Group Health Products:", groupHealthProducts.length);
}

// Call the function to log Group Health products
logGroupHealth();
```



```
    console.log("Collection of Group Health Products:");
    groupHealthProducts.forEach((product, index) => {
        console.log(`${index + 1}. ${product}`);
    });

    // Log the total number of Group Health products
    console.log("Total Group Health Products:", groupHealthProducts.length);
}

// Call the function to log Group Health products
logGroupHealth();
Collection of Group Health Products:                                    VM14462:30
1. Employee Group Health Insurance                                      VM14462:32
2. Group Personal Accident                                              VM14462:32
3. Group Term Life                                                      VM14462:32
4. COVID-19 Group Health Plan                                           VM14462:32
Total Group Health Products: 4                                          VM14462:36
```

## 3. Liability

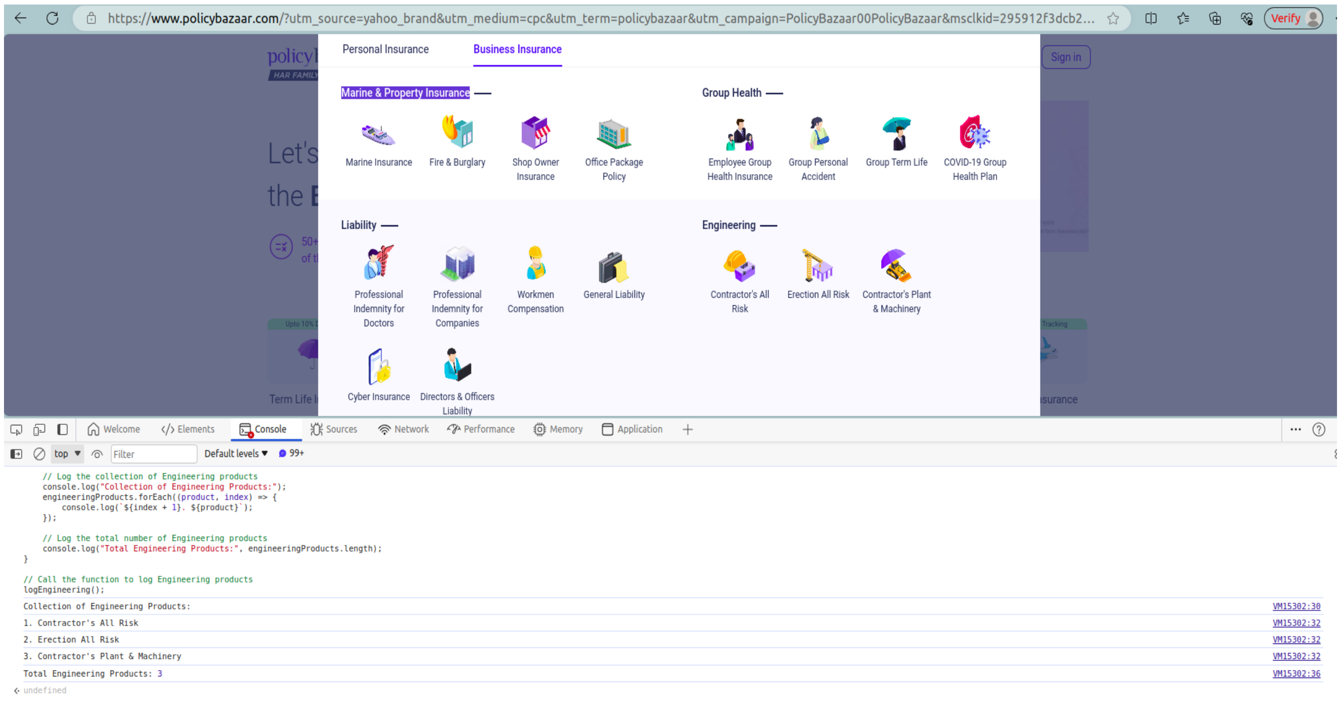
```
// Function to extract Liability products using the provided XPath expression
function extractLiability() {
    // Array to store Liability products
    const liabilityProducts = [];
    // Evaluate the XPath expression to select relevant elements
    const liabilityElements = document.evaluate(
        '//div[@id="tab-content2"]//child::div[2]/child::div[1]/child::ul[1]/li',
        document,
        null,
        XPathResult.ORDERED_NODE_SNAPSHOT_TYPE,
        null
    );

    // Iterate through selected elements and push their text content into the array
    for (let i = 0; i < liabilityElements.snapshotLength; i++) {
        const element = liabilityElements.snapshotItem(i);
        liabilityProducts.push(element.innerText.trim());
    }

    // Return the array of Liability products
    return liabilityProducts;
}

// Function to log the collection of Liability products
function logLiability() {
    // Extract Liability products
    const liabilityProducts = extractLiability();

    // Log the collection of Liability products
    console.log("Collection of Liability Products:");
    liabilityProducts.forEach((product, index) => {
        console.log(`${index + 1}. ${product}`);
    });

    // Log the total number of Liability products
    console.log("Total Liability Products:", liabilityProducts.length);
}

// Call the function to log Liability products
logLiability();
```


https://www.policybazaar.com/?utm_source=yahoo_brand&utm_medium=cpc&utm_term=policybazaar&utm_campaign=PolicyBazaar00PolicyBazaar&msclkid=295912f3dcb2...
Personal Insurance
Business Insurance
Marine & Property Insurance
Marine Insurance
Fire & Burglary
Shop Owner Insurance
Office Package Policy
Group Health
Employee Group Health Insurance
Group Personal Accident
Group Term Life
COVID-19 Group Health Plan
Liability
Professional Indemnity for Doctors
Professional Indemnity for Companies
Workmen Compensation
General Liability
Cyber Insurance
Directors & Officers Liability
Engineering
Contractor's All Risk
Erection All Risk
Contractor's Plant & Machinery
Welcome
Elements
Console
Sources
Network
Performance
Memory
Application
top
Filter
Default levels
99+

```
    // Log the total number of Liability products
    console.log("Total Liability Products:", liabilityProducts.length);
}

// Call the function to log Liability products
logLiability();
Collection of Liability Products:                                   VM15068:30
1. Professional Indemnity for Doctors                               VM15068:32
2. Professional Indemnity for Companies                             VM15068:32
3. Workmen Compensation                                             VM15068:32
4. General Liability                                                VM15068:32
5. Cyber Insurance                                                  VM15068:32
6. Directors & Officers Liability                                   VM15068:32
Total Liability Products: 6                                         VM15068:36
```

## 4. Engineering

```
// Function to extract Engineering products using the provided XPath expression
function extractEngineering() {
    // Array to store Engineering products
    const engineeringProducts = [];
    // Evaluate the XPath expression to select relevant elements
    const engineeringElements = document.evaluate(
        '//div[@id="tab-content2"]//child::div[2]/child::div[2]/child::ul[1]/li',
        document,
        null,
        XPathResult.ORDERED_NODE_SNAPSHOT_TYPE,
        null
    );

    // Iterate through selected elements and push their text content into the array
    for (let i = 0; i < engineeringElements.snapshotLength; i++) {
        const element = engineeringElements.snapshotItem(i);
        engineeringProducts.push(element.innerText.trim());
    }

    // Return the array of Engineering products
    return engineeringProducts;
}

// Function to log the collection of Engineering products
function logEngineering() {
    // Extract Engineering products
    const engineeringProducts = extractEngineering();

    // Log the collection of Engineering products
    console.log("Collection of Engineering Products:");
    engineeringProducts.forEach((product, index) => {
        console.log(`${index + 1}. ${product}`);
    });
    // Log the total number of Engineering products
    console.log("Total Engineering Products:", engineeringProducts.length);
}


// Call the function to log Engineering products
logEngineering();
```



```
    // Log the collection of Engineering products
    console.log("Collection of Engineering Products:");
    engineeringProducts.forEach((product, index) => {
        console.log(`${index + 1}. ${product}`);
    });

    // Log the total number of Engineering products
    console.log("Total Engineering Products:", engineeringProducts.length);
}

// Call the function to log Engineering products
logEngineering();
Collection of Engineering Products:
1. Contractor's All Risk
2. Erection All Risk
3. Contractor's Plant & Machinery
Total Engineering Products: 3
< undefined
```

## Explanation:

- **xpathresult.ordered_node_snapshot_type:** This gets a list of nodes from an XPath query. The nodes are in the same order as they appear in the document.

- **snapshotItem(i):** This is a function that lets we access a specific node in the list by its index i.

- **snapshotLength:** This tells us how many nodes are in the list.

So, if we run an XPath query and use **ordered_node_snapshot_type,** we get a list of nodes. We can find out how many nodes are in the list using **snapshotLength,** and access each node using **snapshotItem(i).** If the document changes, our list stays the same because it's a "snapshot".