PolicyBazaar Home Page Locator Design & Testing

Task 1:

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

Solution:
This task can be performed in many ways I have tried three of them, which are-

1. By ‘windows.open()’ method that opens the provided link in new tab
   window.open('https://www.policybazaar.com/');

➔ the window object represents the browser window or tab that contains the document

➔ open() is a method of the window object. It is used to open a new browser window or tab.

➔ The argument passed to open() is the URL of the page that you want to open in the new window or tab.

2. By using windows properties to navigate to the desired link on the current (same) tab
   window.location.href = 'https://www.policybazaar.com/';

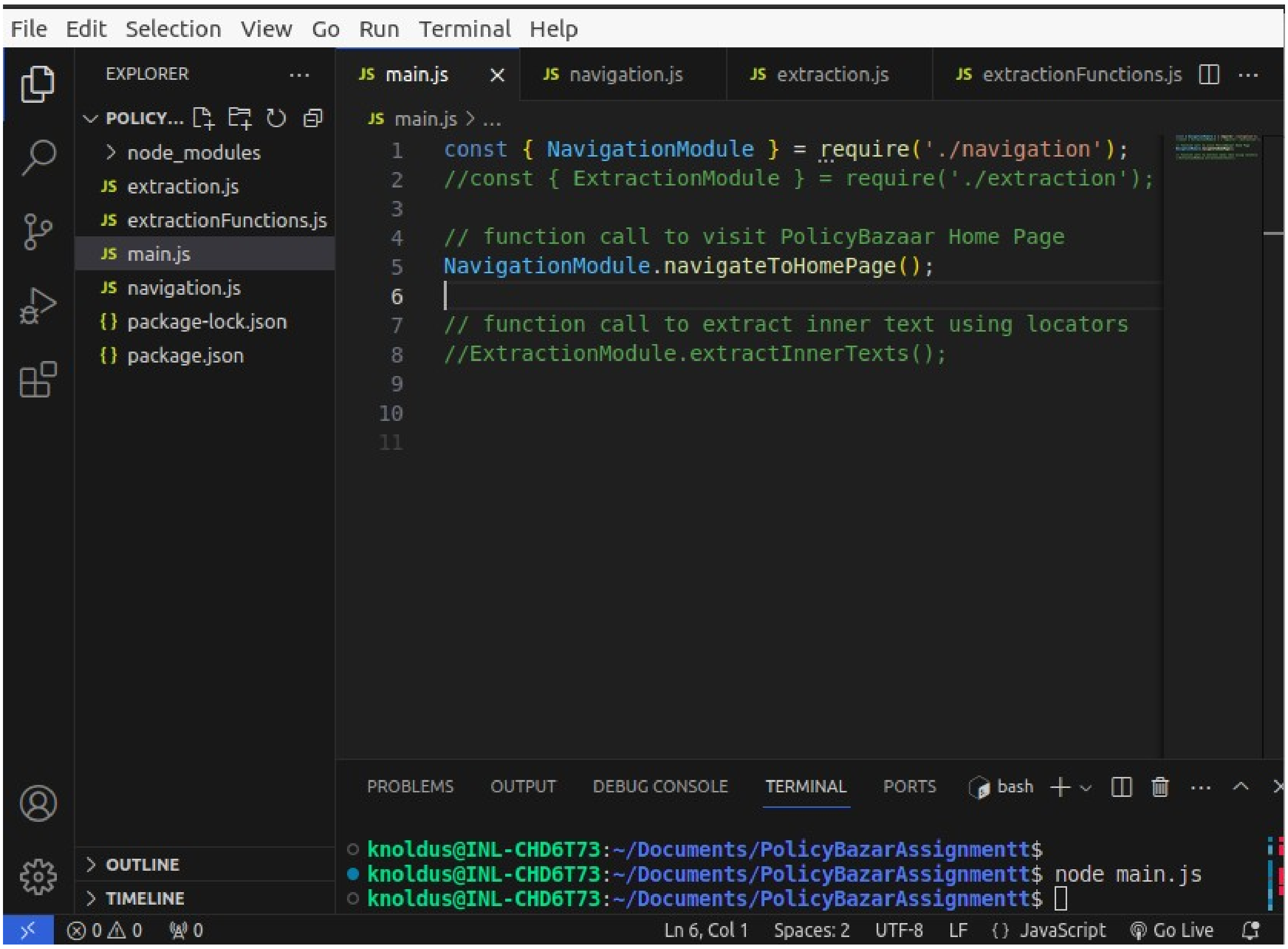
➔ window is a global object in the browser's JavaScript environment. It represents the window or tab containing the webpage.

➔ location is a property of the window object. It provides information about the current URL of the browser window.

➔ href is a property of the location object. It contains the complete URL of the current webpage, including the protocol (http/https), domain, path,etc.

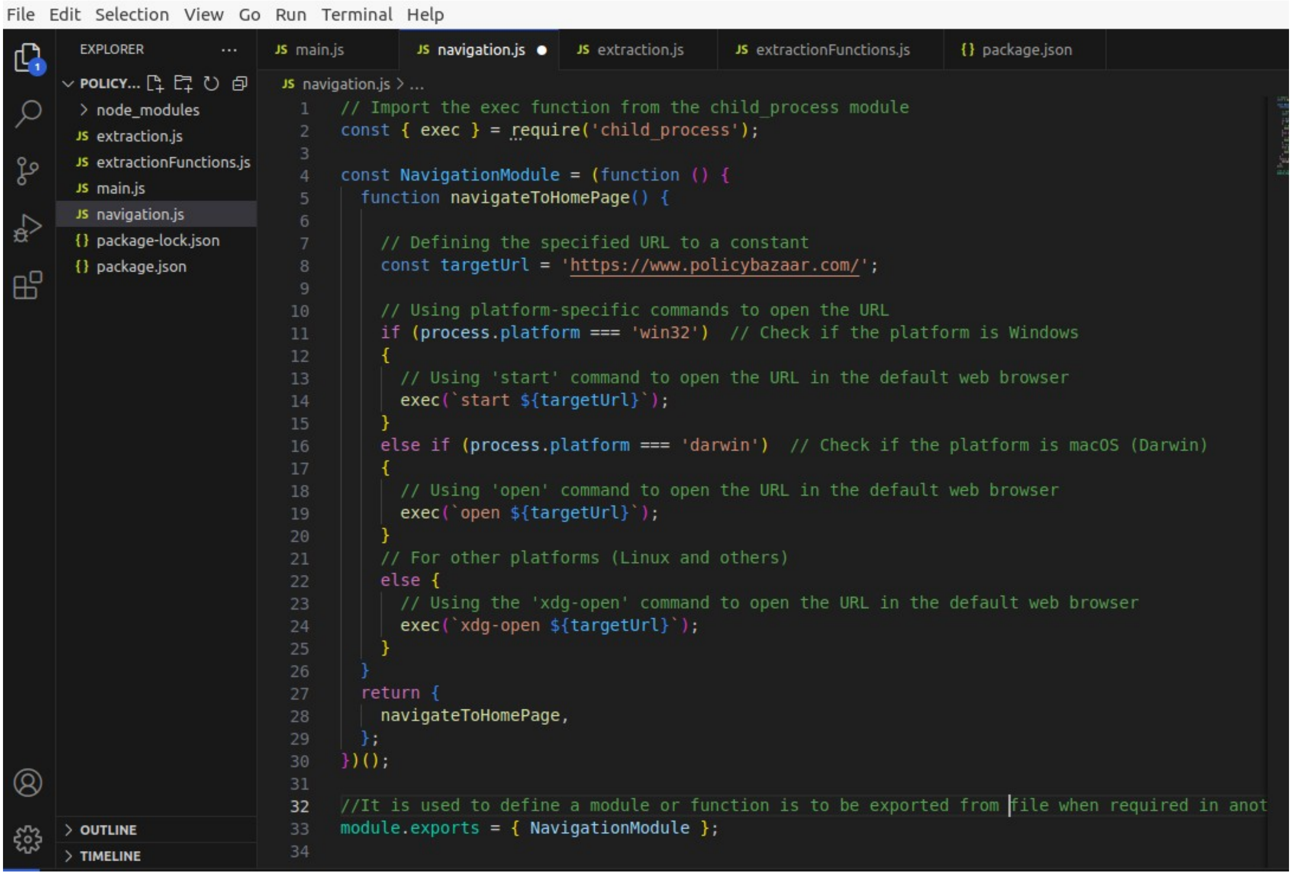
➔ When we set a new value to location.href, we are telling the browser to change the current URL to 'specified url' and load that webpage.

3. By using IDE (VS Code) and execute script through node



```
const { NavigationModule } = require('./navigation');
//const { ExtractionModule } = require('./extraction');

// function call to visit PolicyBazaar Home Page
NavigationModule.navigateToHomePage();

// function call to extract inner text using locators
//ExtractionModule.extractInnerTexts();
```

```
knoldus@INL-CHD6T73:~/Documents/PolicyBazarAssignmentt$
knoldus@INL-CHD6T73:~/Documents/PolicyBazarAssignmentt$ node main.js
knoldus@INL-CHD6T73:~/Documents/PolicyBazarAssignmentt$
```



```
// Import the exec function from the child_process module
const { exec } = require('child_process');

const NavigationModule = (function () {
  function navigateToHomePage() {

    // Defining the specified URL to a constant
    const targetUrl = 'https://www.policybazaar.com/';

    // Using platform-specific commands to open the URL
    if (process.platform === 'win32')  // Check if the platform is Windows
    {
      // Using 'start' command to open the URL in the default web browser
      exec(`start ${targetUrl}`);
    }
    else if (process.platform === 'darwin')  // Check if the platform is macOS (Darwin)
    {
      // Using 'open' command to open the URL in the default web browser
      exec(`open ${targetUrl}`);
    }
    // For other platforms (Linux and others)
    else {
      // Using the 'xdg-open' command to open the URL in the default web browser
      exec(`xdg-open ${targetUrl}`);
    }
  }
  return {
    navigateToHomePage,
  };
})();

//It is used to define a module or function is to be exported from file when required in anot
module.exports = { NavigationModule };
```

Task 2:

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

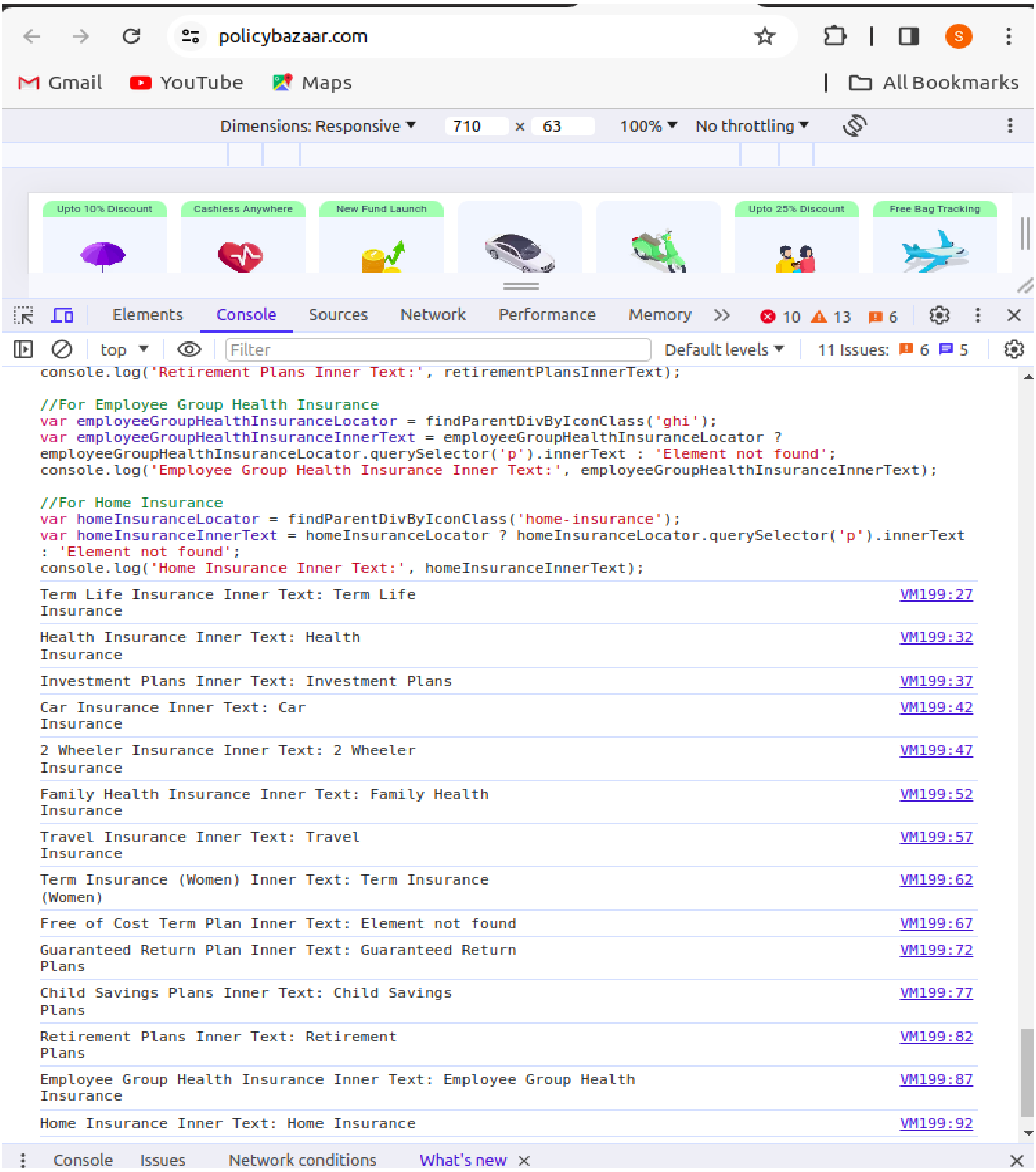
Solution:

```
// Function to find the parent div based on the unique class of the i element
function findParentDivByIconClass(iconClass) {
   // Select all elements with the specified class (outer div’s class)
   var divElements = document.querySelectorAll('.prd-icon.add.shadowHandler.short');

   for (var i = 0; i < divElements.length; i++) {
      // Finding the child i element within the current div element
      var iconElement = divElements[i].querySelector('.shadowHandlerBox i');

      // Checking if the i element has the specified icon class
      if (iconElement && iconElement.classList.contains(iconClass)) {
         // Returning the parent div if the icon class is found
         return divElements[i];
      }
   }

   // Returning null if no matching element is found
   return null;
}

//For Term Life Insurance
// Finding the parent div for 'term-life' icon class
var termLifeInsuranceLocator = findParentDivByIconClass('term-life');
// Extracting the inner text of the first 'p' element within the found div, or setting it to 'Element not found'
var termLifeInnerText = termLifeInsuranceLocator ?
termLifeInsuranceLocator.querySelector('p').innerText : 'Element not found';
// Printing the result to the console
console.log('Term Life Insurance Inner Text:', termLifeInnerText);

//For Health Insurance
var healthInsuranceLocator = findParentDivByIconClass('health-insurance');
var healthInsuranceInnerText = healthInsuranceLocator ?
healthInsuranceLocator.querySelector('p').innerText : 'Element not found';
console.log('Health Insurance Inner Text:', healthInsuranceInnerText);

//For Investment Plans
var investmentPlansLocator = findParentDivByIconClass('investment');
var investmentInnerText = investmentPlansLocator ?
investmentPlansLocator.querySelector('p').innerText : 'Element not found';
console.log('Investment Plans Inner Text:', investmentInnerText);

//For Car Insurance
var carInsuranceLocator = findParentDivByIconClass('car-insurance');
var carInsuranceInnerText = carInsuranceLocator ?
carInsuranceLocator.querySelector('p').innerText : 'Element not found';
```

```
console.log('Car Insurance Inner Text:', carInsuranceInnerText);

//For 2 Wheeler Insurance
var twoWheelerInsuranceLocator = findParentDivByIconClass('two-wheeler-insurance');
var twoWheelerInsuranceInnerText = twoWheelerInsuranceLocator ?
twoWheelerInsuranceLocator.querySelector('p').innerText : 'Element not found';
console.log('2 Wheeler Insurance Inner Text:', twoWheelerInsuranceInnerText);

//For Family Health Insurance
var familyHealthInsuranceLocator = findParentDivByIconClass('family-health-insurance');
var familyHealthInsuranceInnerText = familyHealthInsuranceLocator ?
familyHealthInsuranceLocator.querySelector('p').innerText : 'Element not found';
console.log('Family Health Insurance Inner Text:', familyHealthInsuranceInnerText);

//For Travel Insurance
var travelInsuranceLocator = findParentDivByIconClass('ti');
var travelInsuranceInnerText = travelInsuranceLocator ?
travelInsuranceLocator.querySelector('p').innerText : 'Element not found';
console.log('Travel Insurance Inner Text:', travelInsuranceInnerText);

//For Term Insurance (Women)
var termInsuranceWomenLocator = findParentDivByIconClass('women_term');
var termInsuranceWomenInnerText = termInsuranceWomenLocator ?
termInsuranceWomenLocator.querySelector('p').innerText : 'Element not found';
console.log('Term Insurance (Women) Inner Text:', termInsuranceWomenInnerText);

//For Free of Cost Term Plan
var freeOfCostTermPlanLocator = findParentDivByIconClass('none newterm_icon');
var freeOfCostTermPlanInnerText = freeOfCostTermPlanLocator ?
freeOfCostTermPlanLocator.querySelector('p').innerText : 'Element not found';
console.log('Free of Cost Term Plan Inner Text:', freeOfCostTermPlanInnerText);

//For Guaranteed Return Plan
var guaranteedReturnPlanLocator = findParentDivByIconClass('guaranteed-return-plan');
var guaranteedReturnPlanInnerText = guaranteedReturnPlanLocator ?
guaranteedReturnPlanLocator.querySelector('p').innerText : 'Element not found';
console.log('Guaranteed Return Plan Inner Text:', guaranteedReturnPlanInnerText);

//For Child Savings Plans
var childSavingsPlansLocator = findParentDivByIconClass('csip');
var childSavingsPlansInnerText = childSavingsPlansLocator ?
childSavingsPlansLocator.querySelector('p').innerText : 'Element not found';
console.log('Child Savings Plans Inner Text:', childSavingsPlansInnerText);

//For Retirement Plans
var retirementPlansLocator = findParentDivByIconClass('retirement-plan');
var retirementPlansInnerText = retirementPlansLocator ?
retirementPlansLocator.querySelector('p').innerText : 'Element not found';
console.log('Retirement Plans Inner Text:', retirementPlansInnerText);

//For Employee Group Health Insurance
var employeeGroupHealthInsuranceLocator = findParentDivByIconClass('ghi');
```

var employeeGroupHealthInsuranceInnerText = employeeGroupHealthInsuranceLocator ? employeeGroupHealthInsuranceLocator.querySelector('p').innerText : 'Element not found';
console.log('Employee Group Health Insurance Inner Text:', employeeGroupHealthInsuranceInnerText);

//For Home Insurance
var homeInsuranceLocator = findParentDivByIconClass('home-insurance');
var homeInsuranceInnerText = homeInsuranceLocator ? homeInsuranceLocator.querySelector('p').innerText : 'Element not found';
console.log('Home Insurance Inner Text:', homeInsuranceInnerText);



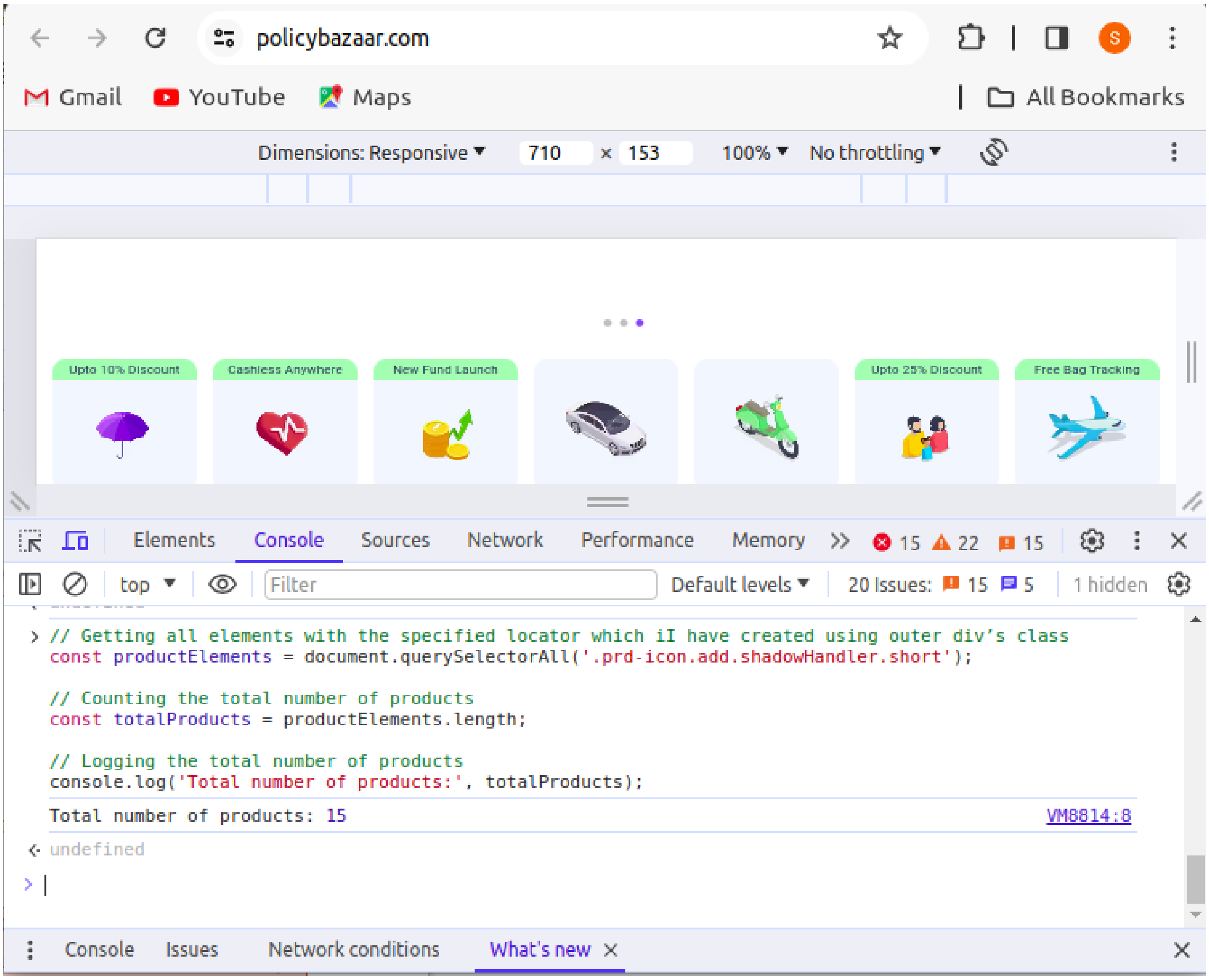
```
console.log('Retirement Plans Inner Text:', retirementPlansInnerText);

//For Employee Group Health Insurance
var employeeGroupHealthInsuranceLocator = findParentDivByIconClass('ghi');
var employeeGroupHealthInsuranceInnerText = employeeGroupHealthInsuranceLocator ?
employeeGroupHealthInsuranceLocator.querySelector('p').innerText : 'Element not found';
console.log('Employee Group Health Insurance Inner Text:', employeeGroupHealthInsuranceInnerText);

//For Home Insurance
var homeInsuranceLocator = findParentDivByIconClass('home-insurance');
var homeInsuranceInnerText = homeInsuranceLocator ? homeInsuranceLocator.querySelector('p').innerText
: 'Element not found';
console.log('Home Insurance Inner Text:', homeInsuranceInnerText);
```

```
Term Life Insurance Inner Text: Term Life                                 VM199:27
Insurance
Health Insurance Inner Text: Health                                       VM199:32
Insurance
Investment Plans Inner Text: Investment Plans                             VM199:37
Car Insurance Inner Text: Car                                             VM199:42
Insurance
2 Wheeler Insurance Inner Text: 2 Wheeler                                 VM199:47
Insurance
Family Health Insurance Inner Text: Family Health                         VM199:52
Insurance
Travel Insurance Inner Text: Travel                                       VM199:57
Insurance
Term Insurance (Women) Inner Text: Term Insurance                         VM199:62
(Women)
Free of Cost Term Plan Inner Text: Element not found                      VM199:67
Guaranteed Return Plan Inner Text: Guaranteed Return                      VM199:72
Plans
Child Savings Plans Inner Text: Child Savings                             VM199:77
Plans
Retirement Plans Inner Text: Retirement                                   VM199:82
Plans
Employee Group Health Insurance Inner Text: Employee Group Health         VM199:87
Insurance
Home Insurance Inner Text: Home Insurance                                 VM199:92
```

- **Count and log the total number of products displayed on the home page.**

Solution:

```
// Getting all elements with the specified locator which iI have created using outer div’s class
const productElements = document.querySelectorAll('.prd-icon.add.shadowHandler.short');

// Counting the total number of products
const totalProducts = productElements.length;

// Logging the total number of products
console.log('Total number of products:', totalProducts);
```



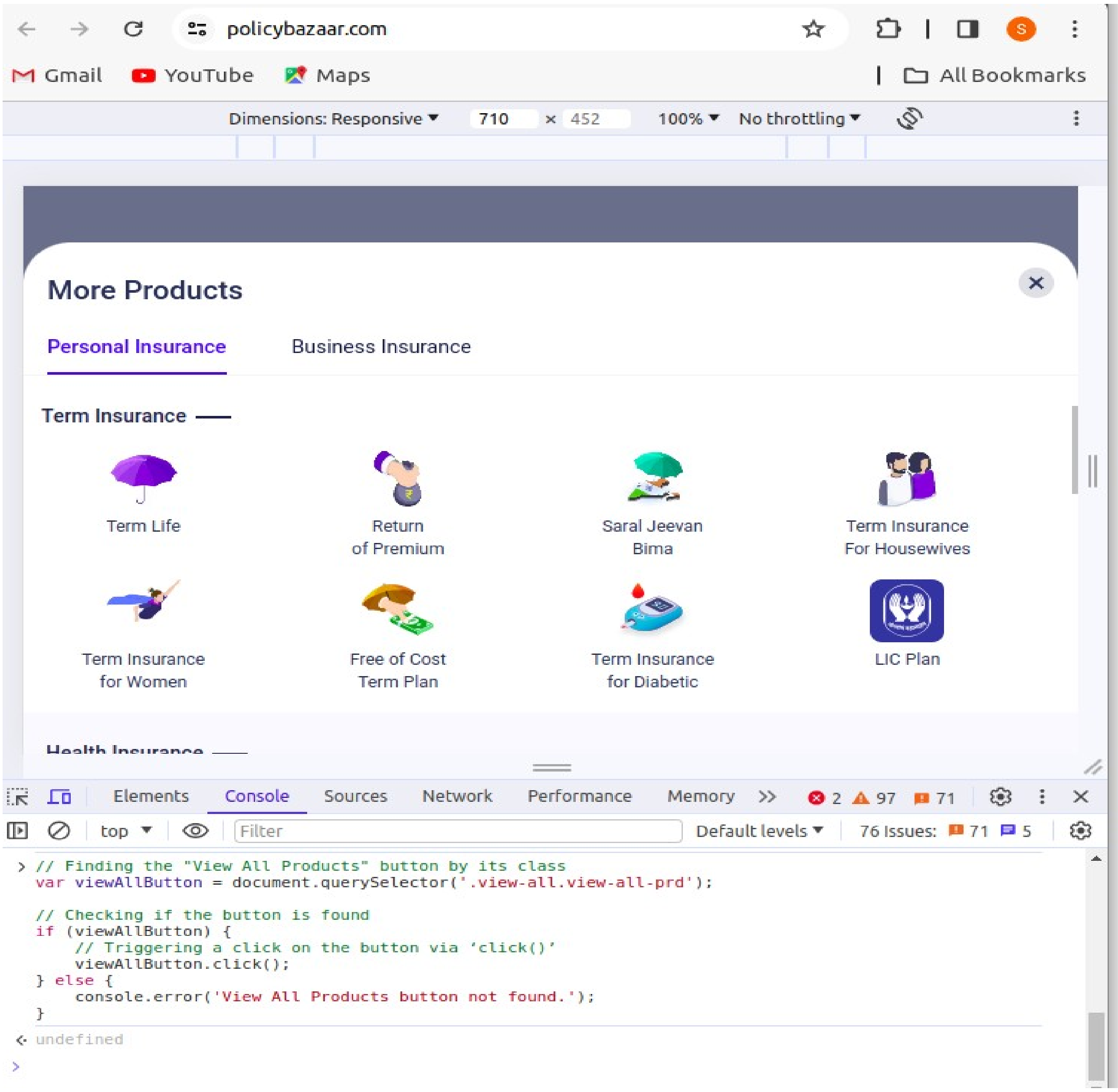
```
> // Getting all elements with the specified locator which iI have created using outer div’s class
  const productElements = document.querySelectorAll('.prd-icon.add.shadowHandler.short');

  // Counting the total number of products
  const totalProducts = productElements.length;

  // Logging the total number of products
  console.log('Total number of products:', totalProducts);
  Total number of products: 15                                                    VM8814:8
<· undefined
>
```

Task 3:

- **Click on "View All Products"**

Solution:

```
// Finding the "View All Products" button by its class
var viewAllButton = document.querySelector('.view-all.view-all-prd');

// Checking if the button is found
if (viewAllButton) {
   // Triggering a click on the button via ‘click()’
   viewAllButton.click();
} else {
   console.error('View All Products button not found.');
}
```



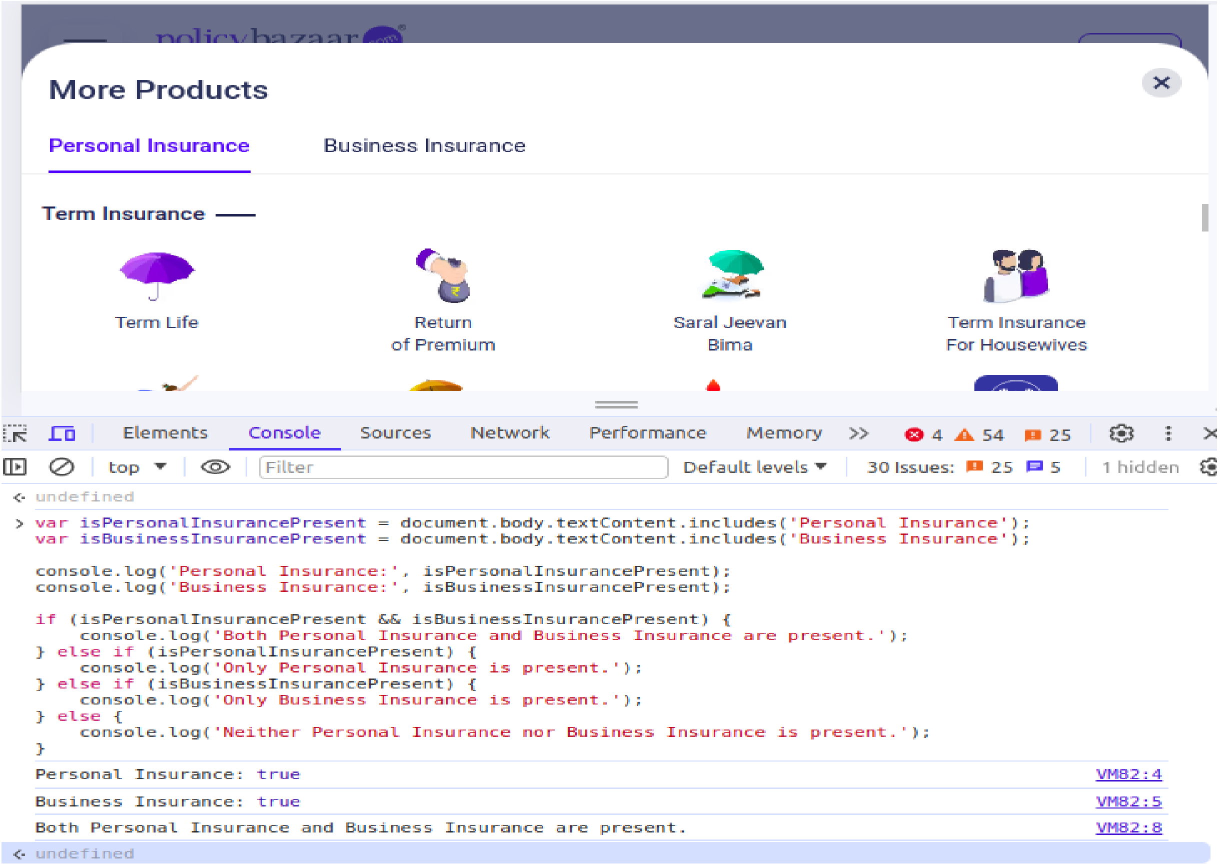
```
> // Finding the "View All Products" button by its class
  var viewAllButton = document.querySelector('.view-all.view-all-prd');

  // Checking if the button is found
  if (viewAllButton) {
      // Triggering a click on the button via ‘click()’
      viewAllButton.click();
  } else {
      console.error('View All Products button not found.');
  }
< undefined
>
```

- **Verify there are two types of insurance products.**

  **** Personal Insurance**
  **** Business Insurance**

Solution:

```
// Checking if "Personal Insurance" is present in the text content of the document body
var isPersonalInsurancePresent = document.body.textContent.includes('Personal Insurance');

// Checking if "Business Insurance" is present in the text content of the document body
var isBusinessInsurancePresent = document.body.textContent.includes('Business Insurance');

// Logging whether Personal Insurance is present
console.log('Personal Insurance:', isPersonalInsurancePresent);

// Logging whether Business Insurance is present
console.log('Business Insurance:', isBusinessInsurancePresent);

// Checking and logging the overall verification results
if (isPersonalInsurancePresent && isBusinessInsurancePresent) {
   console.log('Both Personal Insurance and Business Insurance are present.');
} else if (isPersonalInsurancePresent) {
   console.log('Only Personal Insurance is present.');
} else if (isBusinessInsurancePresent) {
   console.log('Only Business Insurance is present.');
} else {
   console.log('Neither Personal Insurance nor Business Insurance is present.');}
```



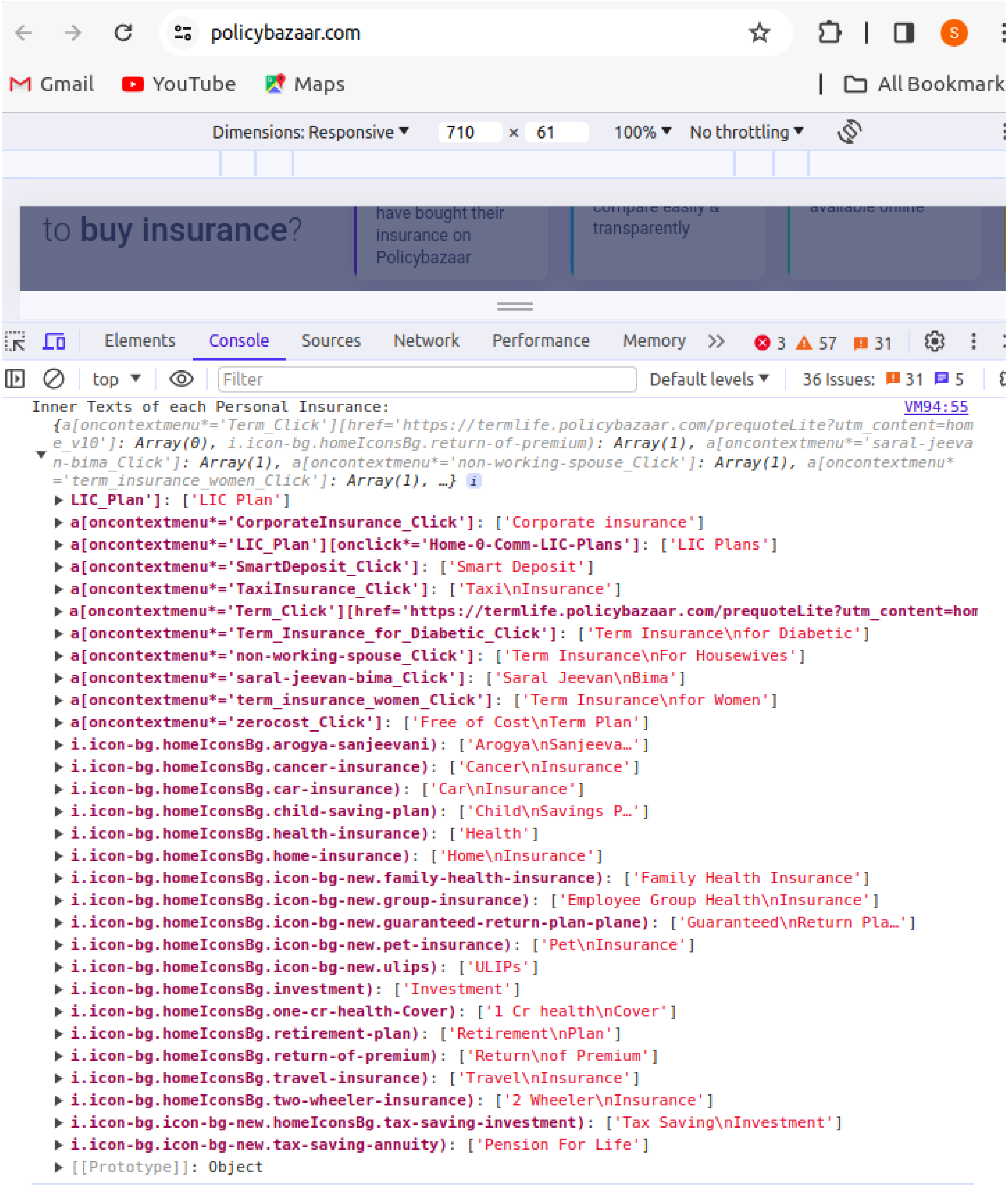
```
<· undefined
> var isPersonalInsurancePresent = document.body.textContent.includes('Personal Insurance');
  var isBusinessInsurancePresent = document.body.textContent.includes('Business Insurance');

  console.log('Personal Insurance:', isPersonalInsurancePresent);
  console.log('Business Insurance:', isBusinessInsurancePresent);

  if (isPersonalInsurancePresent && isBusinessInsurancePresent) {
      console.log('Both Personal Insurance and Business Insurance are present.');
  } else if (isPersonalInsurancePresent) {
      console.log('Only Personal Insurance is present.');
  } else if (isBusinessInsurancePresent) {
      console.log('Only Business Insurance is present.');
  } else {
      console.log('Neither Personal Insurance nor Business Insurance is present.');
  }
  Personal Insurance: true                                                    VM82:4
  Business Insurance: true                                                    VM82:5
  Both Personal Insurance and Business Insurance are present.                 VM82:8
<· undefined
```

Task 4:

- **Design locators to extract inner text information for each personal insurance product.**

Solution:

```
// Defining a function to extract inner text for a given locator
function extractInnerText(locator) {
    // Using document.querySelectorAll to find all elements matching the locator
    const elements = document.querySelectorAll(locator);
    // Converting the NodeList of elements to an array and trim the inner text of each element
    return Array.from(elements, element => element.innerText.trim());
}

const locators = [
    // Locator for Term Life Insurance
    "a[oncontextmenu*='Term_Click'][href='https://termlife.policybazaar.com/prequoteLite?utm_content=home_v10']",
    // Locator for Return of Premium Insurance
    "a:has(> i.icon-bg.homeIconsBg.return-of-premium)",
    // Locator for Saral Jeevan Bima
    "a[oncontextmenu*='saral-jeevan-bima_Click']",
    // Locator for Non-Working Spouse Plan
    "a[oncontextmenu*='non-working-spouse_Click']",
    // Locator for Term Insurance for Women
    "a[oncontextmenu*='term_insurance_women_Click']",
    // Locator for Zero Cost Insurance
    "a[oncontextmenu*='zerocost_Click']",
    // Locator for Term Insurance for Diabetic
    "a[oncontextmenu*='Term_Insurance_for_Diabetic_Click']",
    // Locator for LIC Plans
    "a[oncontextmenu*='Term LIC_Plan']",
    // Locator for Health Insurance
    "a:has(> i.icon-bg.homeIconsBg.health-insurance)",
    // Locator for One Crore Health Cover
    "a:has(> i.icon-bg.homeIconsBg.one-cr-health-Cover)",
    // Locator for Arogya Sanjeevani Insurance
    "a:has(> i.icon-bg.homeIconsBg.arogya-sanjeevani)",
    // Locator for Family Health Insurance
    "a:has(> i.icon-bg.homeIconsBg.icon-bg-new.family-health-insurance)",
    // Locator for Cancer Insurance
    "a:has(> i.icon-bg.homeIconsBg.cancer-insurance)",
    // Locator for LIC Plans (Alternative)
    "a[oncontextmenu*='LIC_Plan'][onclick*='Home-0-Comm-LIC-Plans']",
    // Locator for Investment Plans
    "a:has(> i.icon-bg.homeIconsBg.investment)",
}
    // Locator for Child Saving Plans
    "a:has(> i.icon-bg.homeIconsBg.child-saving-plan)",
    // Locator for Guaranteed Return Plans
    "a:has(> i.icon-bg.homeIconsBg.icon-bg-new.guaranteed-return-plan-plane)",
    // Locator for Retirement Plans
    "a:has(> i.icon-bg.homeIconsBg.retirement-plan)",
    // Locator for Tax Saving Investment
    "a:has(> i.icon-bg.icon-bg-new.homeIconsBg.tax-saving-investment)",
    // Locator for Tax Saving Annuity
    "a:has(> i.icon-bg.icon-bg-new.tax-saving-annuity)",
    // Locator for Smart Deposit
    "a[oncontextmenu*='SmartDeposit_Click']",
    // Locator for ULIPs
    "a:has(> i.icon-bg.homeIconsBg.icon-bg-new.ulips)",
    // Locator for Car Insurance
    "a:has(> i.icon-bg.homeIconsBg.car-insurance)",
    // Locator for Two Wheeler Insurance
    "a:has(> i.icon-bg.homeIconsBg.two-wheeler-insurance)",
    // Locator for Travel Insurance
    "a:has(> i.icon-bg.homeIconsBg.travel-insurance)",
    // Locator for Pet Insurance
    "a:has(> i.icon-bg.homeIconsBg.icon-bg-new.pet-insurance)",
    // Locator for Home Insurance
    "a:has(> i.icon-bg.homeIconsBg.home-insurance)",
    // Locator for Taxi Insurance
    "a[oncontextmenu*='TaxiInsurance_Click']",
    // Locator for Group Insurance
    "a:has(> i.icon-bg.homeIconsBg.icon-bg-new.group-insurance)",
    // Locator for Corporate Insurance
    "a[oncontextmenu*='CorporateInsurance_Click']"
];

// Creating an object to store inner texts for each product
const innerTexts = {};

// Extracting inner text for each product using loop
for (const locator of locators) {
    // Extracting product name from the locator
    const productName = locator.split(' ').pop();
    // Call the extractInnerText function to get inner text for the locator
    innerTexts[productName] = extractInnerText(locator);
}

// Log the inner texts for each product
console.log("Inner Texts of Elements Identified by Locators:", innerTexts);
```



```
Inner Texts of each Personal Insurance:                                                      VM94:55
  {a[oncontextmenu*='Term_Click'][href='https://termlife.policybazaar.com/prequoteLite?utm_content=hom
  e_v10']: Array(0), i.icon-bg.homeIconsBg.return-of-premium): Array(1), a[oncontextmenu*='saral-jeeva
▼ n-bima_Click']: Array(1), a[oncontextmenu*='non-working-spouse_Click']: Array(1), a[oncontextmenu*
  ='term_insurance_women_Click']: Array(1), …} i
  ▶ LIC_Plan']: ['LIC Plan']
  ▶ a[oncontextmenu*='CorporateInsurance_Click']: ['Corporate insurance']
  ▶ a[oncontextmenu*='LIC_Plan'][onclick*='Home-0-Comm-LIC-Plans']: ['LIC Plans']
  ▶ a[oncontextmenu*='SmartDeposit_Click']: ['Smart Deposit']
  ▶ a[oncontextmenu*='TaxiInsurance_Click']: ['Taxi\nInsurance']
  ▶ a[oncontextmenu*='Term_Click'][href='https://termlife.policybazaar.com/prequoteLite?utm_content=hom
  ▶ a[oncontextmenu*='Term_Insurance_for_Diabetic_Click']: ['Term Insurance\nfor Diabetic']
  ▶ a[oncontextmenu*='non-working-spouse_Click']: ['Term Insurance\nFor Housewives']
  ▶ a[oncontextmenu*='saral-jeevan-bima_Click']: ['Saral Jeevan\nBima']
  ▶ a[oncontextmenu*='term_insurance_women_Click']: ['Term Insurance\nfor Women']
  ▶ a[oncontextmenu*='zerocost_Click']: ['Free of Cost\nTerm Plan']
  ▶ i.icon-bg.homeIconsBg.arogya-sanjeevani): ['Arogya\nSanjeeva…']
  ▶ i.icon-bg.homeIconsBg.cancer-insurance): ['Cancer\nInsurance']
  ▶ i.icon-bg.homeIconsBg.car-insurance): ['Car\nInsurance']
  ▶ i.icon-bg.homeIconsBg.child-saving-plan): ['Child\nSavings P…']
  ▶ i.icon-bg.homeIconsBg.health-insurance): ['Health']
  ▶ i.icon-bg.homeIconsBg.home-insurance): ['Home\nInsurance']
  ▶ i.icon-bg.homeIconsBg.icon-bg-new.family-health-insurance): ['Family Health Insurance']
  ▶ i.icon-bg.homeIconsBg.icon-bg-new.group-insurance): ['Employee Group Health\nInsurance']
  ▶ i.icon-bg.homeIconsBg.icon-bg-new.guaranteed-return-plan-plane): ['Guaranteed\nReturn Pla…']
  ▶ i.icon-bg.homeIconsBg.icon-bg-new.pet-insurance): ['Pet\nInsurance']
  ▶ i.icon-bg.homeIconsBg.icon-bg-new.ulips): ['ULIPs']
  ▶ i.icon-bg.homeIconsBg.investment): ['Investment']
  ▶ i.icon-bg.homeIconsBg.one-cr-health-Cover): ['1 Cr health\nCover']
  ▶ i.icon-bg.homeIconsBg.retirement-plan): ['Retirement\nPlan']
  ▶ i.icon-bg.homeIconsBg.return-of-premium): ['Return\nof Premium']
  ▶ i.icon-bg.homeIconsBg.travel-insurance): ['Travel\nInsurance']
  ▶ i.icon-bg.homeIconsBg.two-wheeler-insurance): ['2 Wheeler\nInsurance']
  ▶ i.icon-bg.icon-bg-new.homeIconsBg.tax-saving-investment): ['Tax Saving\nInvestment']
  ▶ i.icon-bg.icon-bg-new.tax-saving-annuity): ['Pension For Life']
  ▶ [[Prototype]]: Object
```

- **Log the collection of personal insurance products along with their counts.**

Solution:

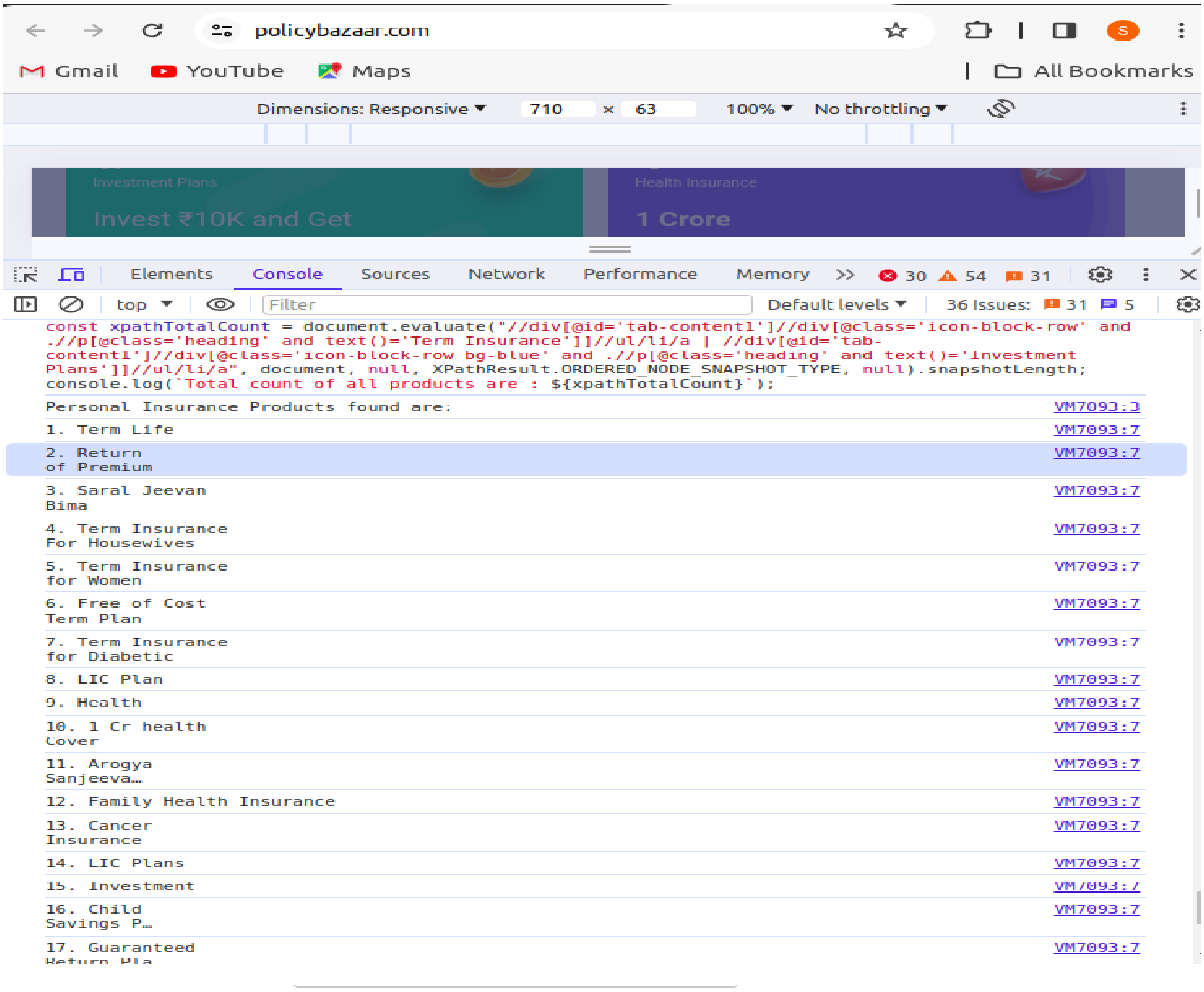
```
// Function to count and list links based on the provided XPath
function countAndListLinks(categoryHeading, xpath) {
 // Evaluate the XPath expression and get the result
   const xpathResult = document.evaluate(xpath, document, null,
XPathResult.ORDERED_NODE_SNAPSHOT_TYPE, null);

   // Logging a message indicating the category of products found
   console.log(`Personal Insurance Products found are:`);

   // Loop through each result item in the XPath result
   for (let i = 0; i < xpathResult.snapshotLength; i++) {
      const link = xpathResult.snapshotItem(i);

      // Logging the index, trimmed text content of the link
      console.log(`${i + 1}. ${link.innerText.trim()}`);
   }
}

// Count and list Personal Insurance Products links using the XPath
countAndListLinks("Personal Insurance Products", "//div[@id='tab-content1']//div[@class='icon-block-row' and .//p[@class='heading' and text()='Term Insurance']]//ul/li/a | //div[@id='tab-content1']//div[@class='icon-block-row bg-blue' and .//p[@class='heading' and text()='Investment Plans']]//ul/li/a");

// Calculate the total count of all product links
const xpathTotalCount = document.evaluate("//div[@id='tab-content1']//div[@class='icon-block-row' and .//p[@class='heading' and text()='Term Insurance']]//ul/li/a |
//div[@id='tab-content1']//div[@class='icon-block-row bg-blue' and .//p[@class='heading' and
text()='Investment Plans']]//ul/li/a", document, null,
XPathResult.ORDERED_NODE_SNAPSHOT_TYPE, null).snapshotLength;

// Log the total count of all product links
console.log(`Total count of all products are : ${xpathTotalCount}`);
```

Explanation:
A static collection of nodes in the DOM that match the given XPath expression at the moment of the evaluation is referred to as a "**snapshot**" because it takes a picture of the nodes that satisfies the requirements at the time the XPath expression was run.

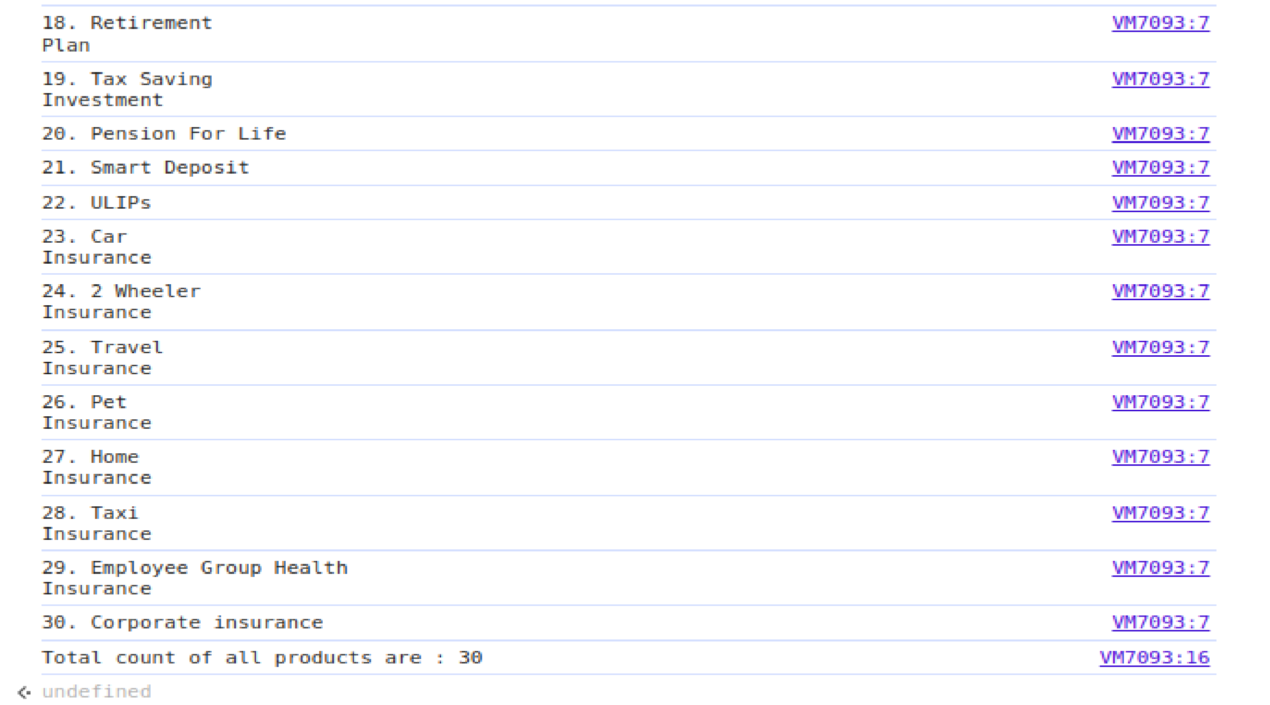
- **XPathResult.ORDERED_NODE_SNAPSHOT_TYPE:** This is a constant indicating that the XPath expression should return a snapshot of nodes in the order in which they appear in the document.

- **snapshotLength:** After the XPath expression is evaluated, snapshotLength is used to determine how many nodes (elements) were found that match the specified criteria.

- **snapshotItem(i):** This method is used in a loop to access individual nodes within the snapshot. It retrieves the node at the specified index (i) in the ordered snapshot.



```
const xpathTotalCount = document.evaluate("//div[@id='tab-content1']//div[@class='icon-block-row' and
.//p[@class='heading' and text()='Term Insurance']]//ul/li/a | //div[@id='tab-
content1']//div[@class='icon-block-row bg-blue' and .//p[@class='heading' and text()='Investment
Plans']]//ul/li/a", document, null, XPathResult.ORDERED_NODE_SNAPSHOT_TYPE, null).snapshotLength;
console.log(`Total count of all products are : ${xpathTotalCount}`);
Personal Insurance Products found are:                                  VM7093:3
1. Term Life                                                            VM7093:7
2. Return                                                               VM7093:7
of Premium
3. Saral Jeevan                                                         VM7093:7
Bima
4. Term Insurance                                                       VM7093:7
For Housewives
5. Term Insurance                                                       VM7093:7
for Women
6. Free of Cost                                                         VM7093:7
Term Plan
7. Term Insurance                                                       VM7093:7
for Diabetic
8. LIC Plan                                                             VM7093:7
9. Health                                                               VM7093:7
10. 1 Cr health                                                         VM7093:7
Cover
11. Arogya                                                              VM7093:7
Sanjeeva…
12. Family Health Insurance                                             VM7093:7
13. Cancer                                                              VM7093:7
Insurance
14. LIC Plans                                                           VM7093:7
15. Investment                                                          VM7093:7
16. Child                                                               VM7093:7
Savings P…
17. Guaranteed                                                          VM7093:7
Return Pla
```



```
18. Retirement                                                          VM7093:7
Plan
19. Tax Saving                                                          VM7093:7
Investment
20. Pension For Life                                                    VM7093:7
21. Smart Deposit                                                       VM7093:7
22. ULIPs                                                               VM7093:7
23. Car                                                                 VM7093:7
Insurance
24. 2 Wheeler                                                           VM7093:7
Insurance
25. Travel                                                              VM7093:7
Insurance
26. Pet                                                                 VM7093:7
Insurance
27. Home                                                                VM7093:7
Insurance
28. Taxi                                                                VM7093:7
Insurance
29. Employee Group Health                                               VM7093:7
Insurance
30. Corporate insurance                                                 VM7093:7
Total count of all products are : 30                                    VM7093:16
← undefined
```

Task 5:

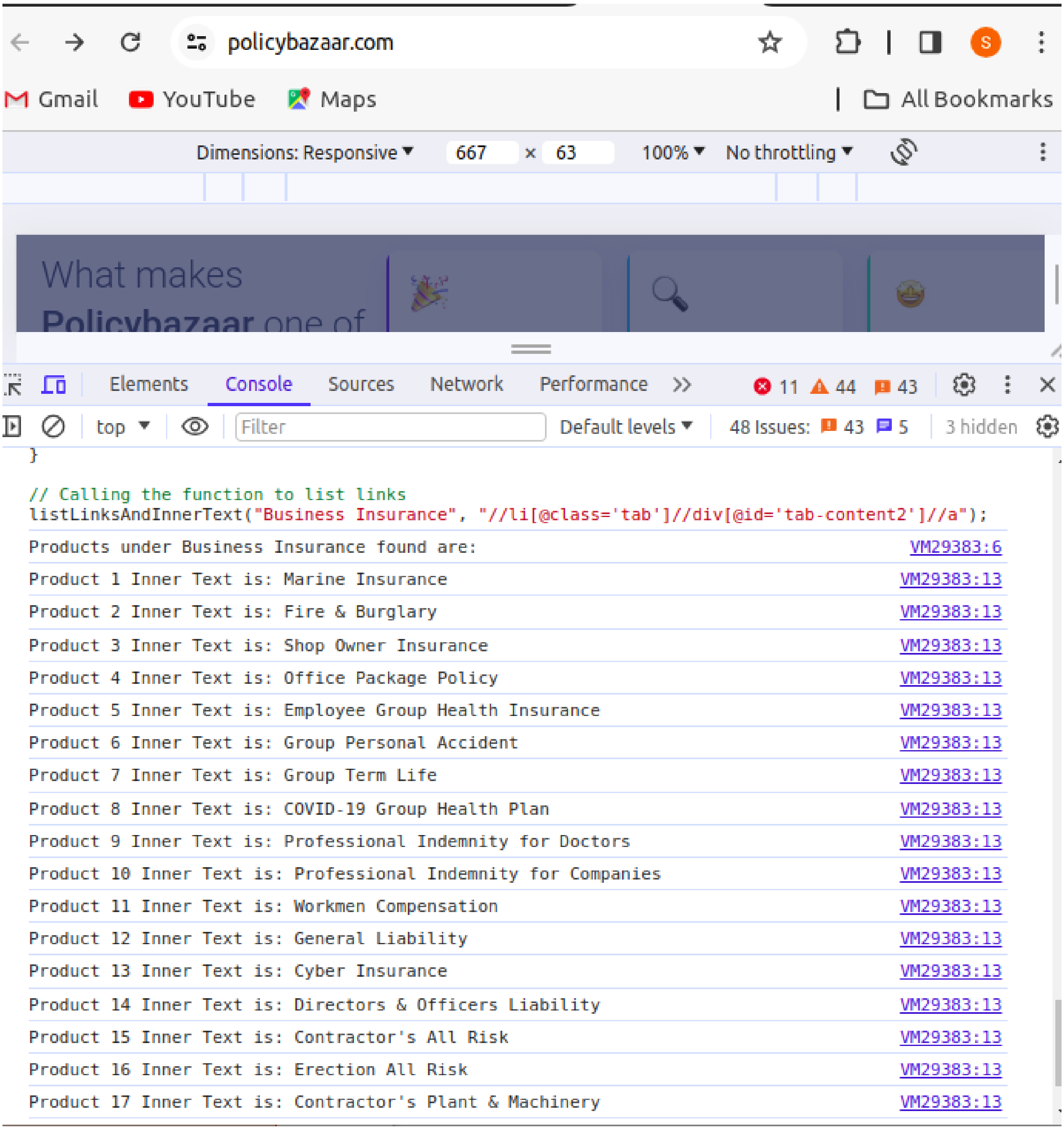
- **Design locators to extract inner text information for each business insurance product.**

Solution:

We can implement this by two approaches i.e designing locators for each product as well as getting all products at once

APPROACH 1: Getting all products at once

```
// Function to list links under their heading
function listLinksAndInnerText(categoryHeading, xpath) {
   // Using XPath to get a snapshot of elements matching the given XPath expression
   const xpathResult = document.evaluate(xpath, document, null,
XPathResult.ORDERED_NODE_SNAPSHOT_TYPE, null);
   // Printing a message indicating the products under the specified category are found
   console.log(`Products under ${categoryHeading} found are:`);


   // Loop through each element in the XPath snapshot
   for (let i = 0; i < xpathResult.snapshotLength; i++) {
      const link = xpathResult.snapshotItem(i);
      const innerText = link.querySelector('span.link').innerText.trim();
      // Printing the information about the current product, including its index and inner text
      console.log(`Product ${i + 1} Inner Text is: ${innerText}`);
   }
   // Printing a newline for better readability
   console.log('\n');
}


// Calling the function to list links
listLinksAndInnerText("Business Insurance", "//li[@class='tab']//div[@id='tab-content2']//a");
```

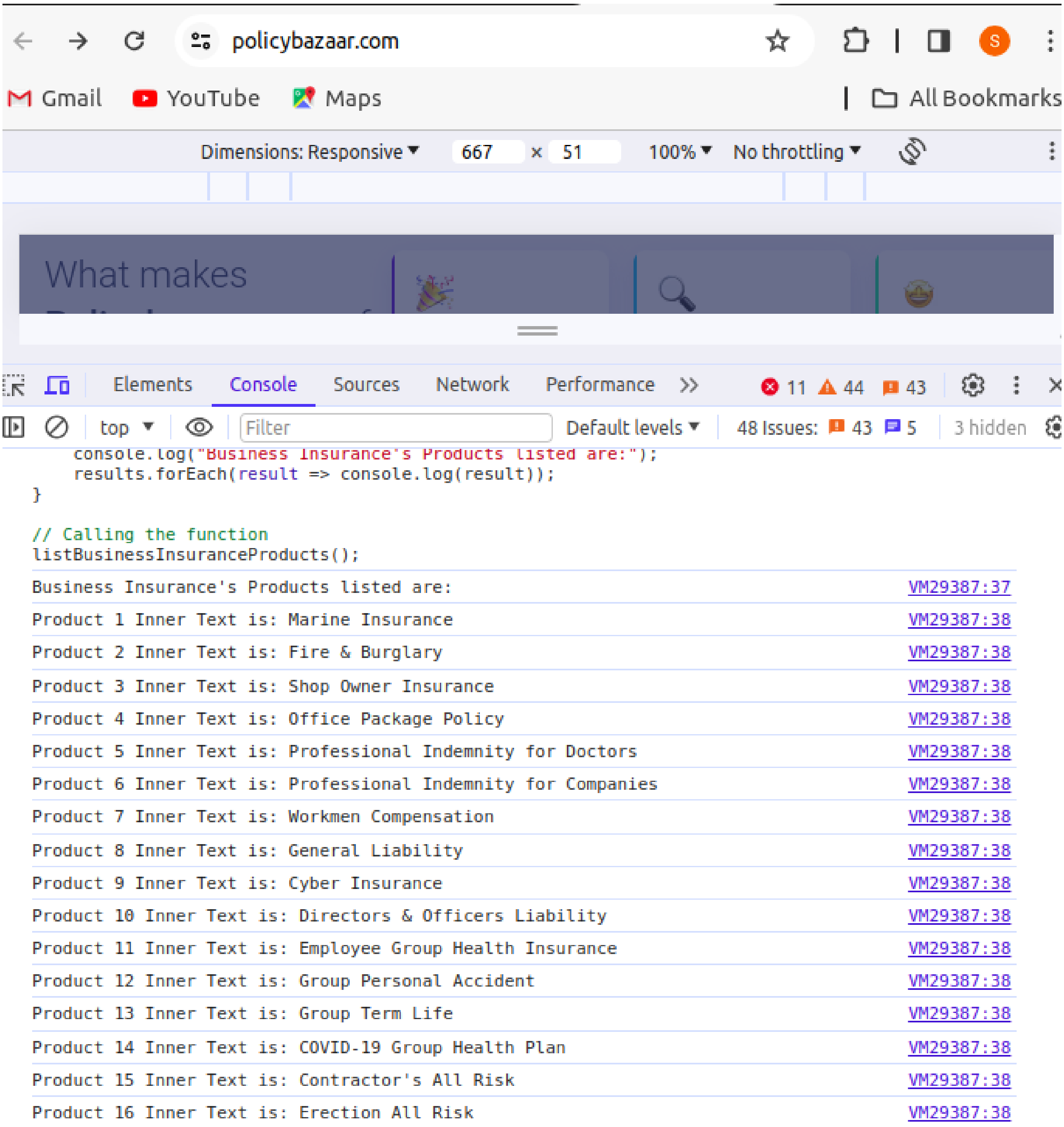


```
}

// Calling the function to list links
listLinksAndInnerText("Business Insurance", "//li[@class='tab']//div[@id='tab-content2']//a");
Products under Business Insurance found are:                              VM29383:6
Product 1 Inner Text is: Marine Insurance                                VM29383:13
Product 2 Inner Text is: Fire & Burglary                                 VM29383:13
Product 3 Inner Text is: Shop Owner Insurance                            VM29383:13
Product 4 Inner Text is: Office Package Policy                           VM29383:13
Product 5 Inner Text is: Employee Group Health Insurance                 VM29383:13
Product 6 Inner Text is: Group Personal Accident                         VM29383:13
Product 7 Inner Text is: Group Term Life                                 VM29383:13
Product 8 Inner Text is: COVID-19 Group Health Plan                      VM29383:13
Product 9 Inner Text is: Professional Indemnity for Doctors              VM29383:13
Product 10 Inner Text is: Professional Indemnity for Companies           VM29383:13
Product 11 Inner Text is: Workmen Compensation                           VM29383:13
Product 12 Inner Text is: General Liability                              VM29383:13
Product 13 Inner Text is: Cyber Insurance                                VM29383:13
Product 14 Inner Text is: Directors & Officers Liability                 VM29383:13
Product 15 Inner Text is: Contractor's All Risk                          VM29383:13
Product 16 Inner Text is: Erection All Risk                              VM29383:13
Product 17 Inner Text is: Contractor's Plant & Machinery                 VM29383:13
```

```
}
```

APPROACH 2: Designing locators for each product

```
// Function to find list for each business insurance product with their inner text
function listBusinessInsuranceProducts() {
    const productLocators = [
        "a[onclick*='MarineInsurance']",
        "a[onclick*='FireandBurglary']",
        "a[onclick*='Shop Owner Insurance']",
        "a[onclick*='Office Package Policy']",
        "a:has(> i.icon-bg.professional-indemnity)",
        "a[onclick*='Professional Indemnity for Companies']",
        "a:has(> i.icon-bg.workmen-compensation)",
        "a:has(> i.icon-bg.general-liability)",
        "a[onclick*='Cyber Insurance']",
        "a:has(> i.icon-bg.directors-officers-liability)",
        "a:has(> i.icon-bg.group-health-insurance)",
        "a[onclick*='Group Personal Accident']",
        "a[onclick*='Group Term Life']",
        "a[onclick*='Group Health Plan']",
        "a:has(> i.icon-bg.construction-all-risk)",
        "a[onclick*='Erection All Risk']",
        "a[onclick*='Contractors Plant & Machinery']"
    ];

    // array to store inner text information
    const results = [];

    // Iterate through each product locator
    productLocators.forEach((locator, index) => {
        const element = document.querySelector(locator);
        if (element) {
            // Extracting inner text information
            const innerText = element.innerText.trim();
            results.push(`Product ${index + 1} Inner Text is: ${innerText}`);
        }
    });

    // Printing the results
    console.log("Business Insurance's Products listed are:");
    results.forEach(result => console.log(result));
}

// Calling the function
listBusinessInsuranceProducts();
```



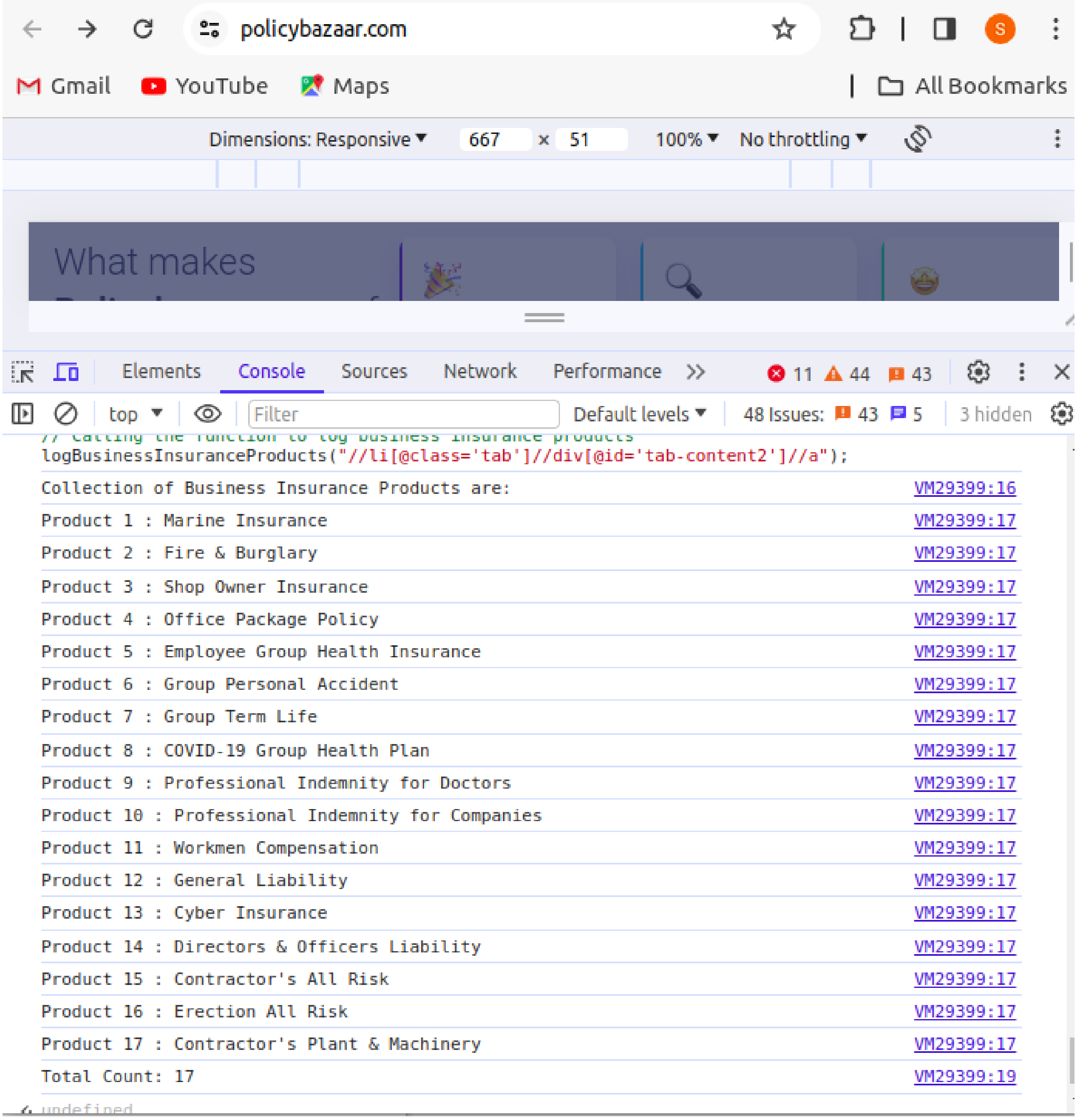
```
    console.log("Business Insurance's Products listed are:");
    results.forEach(result => console.log(result));
}

// Calling the function
listBusinessInsuranceProducts();
```

```
Business Insurance's Products listed are:                                VM29387:37
Product 1 Inner Text is: Marine Insurance                                VM29387:38
Product 2 Inner Text is: Fire & Burglary                                 VM29387:38
Product 3 Inner Text is: Shop Owner Insurance                            VM29387:38
Product 4 Inner Text is: Office Package Policy                           VM29387:38
Product 5 Inner Text is: Professional Indemnity for Doctors              VM29387:38
Product 6 Inner Text is: Professional Indemnity for Companies            VM29387:38
Product 7 Inner Text is: Workmen Compensation                            VM29387:38
Product 8 Inner Text is: General Liability                               VM29387:38
Product 9 Inner Text is: Cyber Insurance                                 VM29387:38
Product 10 Inner Text is: Directors & Officers Liability                 VM29387:38
Product 11 Inner Text is: Employee Group Health Insurance                VM29387:38
Product 12 Inner Text is: Group Personal Accident                        VM29387:38
Product 13 Inner Text is: Group Term Life                                VM29387:38
Product 14 Inner Text is: COVID-19 Group Health Plan                     VM29387:38
Product 15 Inner Text is: Contractor's All Risk                          VM29387:38
Product 16 Inner Text is: Erection All Risk                              VM29387:38
```

- **Log the collection of business insurance products along with their counts.**

Solution:

```
function logBusinessInsuranceProducts(xpath) {
    // Using XPath to get a snapshot of elements matching the given XPath expression
    const xpathResult = document.evaluate(xpath, document, null,
XPathResult.ORDERED_NODE_SNAPSHOT_TYPE, null);
    // Results array to store information about business insurance products
    const results = [];

    // Loop through each element in the XPath snapshot
    for (let i = 0; i < xpathResult.snapshotLength; i++) {
        const link = xpathResult.snapshotItem(i);
        const innerText = link.querySelector('span.link').innerText.trim();
        // Adding information about the current product to the results array
        results.push(`Product ${i + 1} : ${innerText}`);
    }

    // Logging the collection of business insurance products
    console.log("Collection of Business Insurance Products are:");
    results.forEach(result => console.log(result));
    // Logging the total count of business insurance products
    console.log(`Total Count: ${results.length}\n`);
}

// Calling the function to log business insurance products
logBusinessInsuranceProducts("//li[@class='tab']//div[@id='tab-content2']//a");
```



```
// Calling the function to log business insurance products
logBusinessInsuranceProducts("//li[@class='tab']//div[@id='tab-content2']//a");
Collection of Business Insurance Products are:                          VM29399:16
Product 1 : Marine Insurance                                            VM29399:17
Product 2 : Fire & Burglary                                             VM29399:17
Product 3 : Shop Owner Insurance                                        VM29399:17
Product 4 : Office Package Policy                                       VM29399:17
Product 5 : Employee Group Health Insurance                             VM29399:17
Product 6 : Group Personal Accident                                     VM29399:17
Product 7 : Group Term Life                                             VM29399:17
Product 8 : COVID-19 Group Health Plan                                  VM29399:17
Product 9 : Professional Indemnity for Doctors                          VM29399:17
Product 10 : Professional Indemnity for Companies                       VM29399:17
Product 11 : Workmen Compensation                                       VM29399:17
Product 12 : General Liability                                          VM29399:17
Product 13 : Cyber Insurance                                            VM29399:17
Product 14 : Directors & Officers Liability                             VM29399:17
Product 15 : Contractor's All Risk                                      VM29399:17
Product 16 : Erection All Risk                                          VM29399:17
Product 17 : Contractor's Plant & Machinery                             VM29399:17
Total Count: 17                                                         VM29399:19
undefined
```