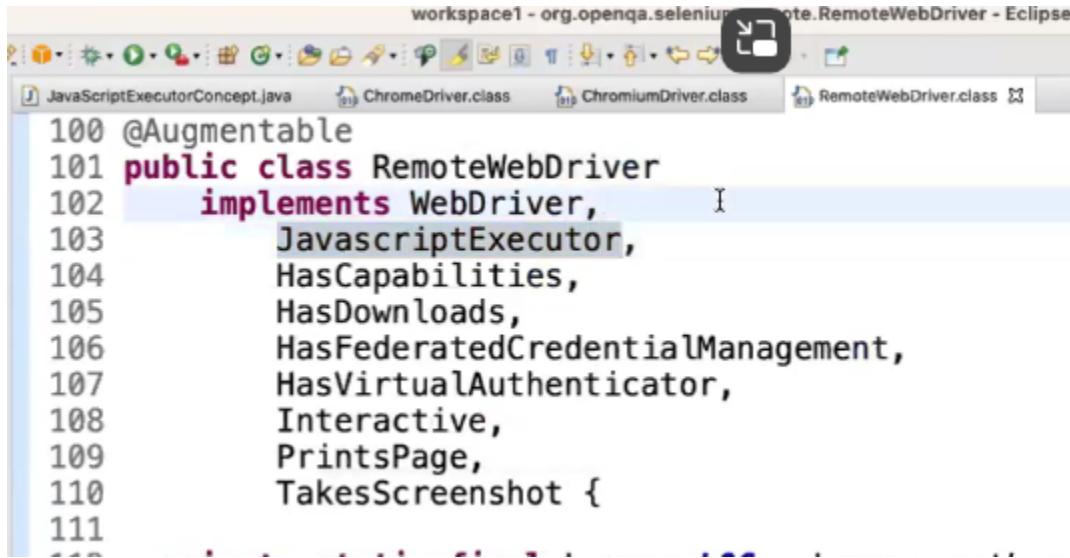


When normal java code doesn't work then use js executor.

Browser understands only js.

Executor will add js into the browser.

Remote web driver implements js executor.



The screenshot shows the Eclipse IDE interface with the title bar "workspace1 - org.openqa.selenium - RemoteWebDriver - Eclipse". Below the title bar, there is a toolbar with various icons. The main workspace contains four tabs: "JavaScriptExecutorConcept.java", "ChromeDriver.class", "ChromiumDriver.class", and "RemoteWebDriver.class". The "RemoteWebDriver.class" tab is currently selected and displays the following Java code:

```
100 @Augmentable
101 public class RemoteWebDriver
102     implements WebDriver, I
103     JavascriptExecutor,
104     HasCapabilities,
105     HasDownloads,
106     HasFederatedCredentialManagement,
107     HasVirtualAuthenticator,
108     Interactive,
109     PrintsPage,
110     TakesScreenshot {
111 }
```

Only one method inside js executor interface.

```

49     * <li>For a non-decimal number, a Long is returned
50     * <li>For a boolean, a Boolean is returned
51     * <li>For all other cases, a String is returned.
52     * <li>For an array, return a List<Object>; with each object following the rules above. We
53     * support nested lists.
54     * <li>For a map, return a Map<String, Object>; with values following the rules above.
55     * <li>Unless the value is null or there is no return value, in which null is returned
56     */
57 
```

58 *

59 * <p>Arguments must be a number, a boolean, a String, WebElement, or a List of any combination of

60 * the above. An exception will be thrown if the arguments do not meet these criteria. The

61 * arguments will be made available to the JavaScript via the "arguments" magic variable, as if

62 * the function were called via "Function.apply"

63 *

64 * @param script The JavaScript to execute

65 * @param args The arguments to the script. May be empty

66 * @return One of Boolean, Long, Double, String, List, Map or WebElement. Or null.

67 */

68 @Nullable Object executeScript(String script, @Nullable Object... args);

69

70 /**
71 * Execute an asynchronous piece of JavaScript in the context of the currently selected frame or
72 * window. Unlike executing {@link #executeScript(String, Object...)} synchronous JavaScript,
73 * scripts executed with this method must explicitly signal they are finished by invoking the
74 * provided callback. This callback is always injected into the executed function as the last
75 * argument.
76 *
77 * <p>The first argument passed to the callback function will be used as the script's result. This
78 * value will be handled as follows:
79 *
80 *
81 * For an HTML element, this method returns a WebElement
82 * For a number, a Long is returned
83 * For a boolean, a Boolean is returned
84 * For all other cases, a String is returned.
85 * For an array, return a List<Object>; with each object following the rules above. We

Remote web driver has implemented it.

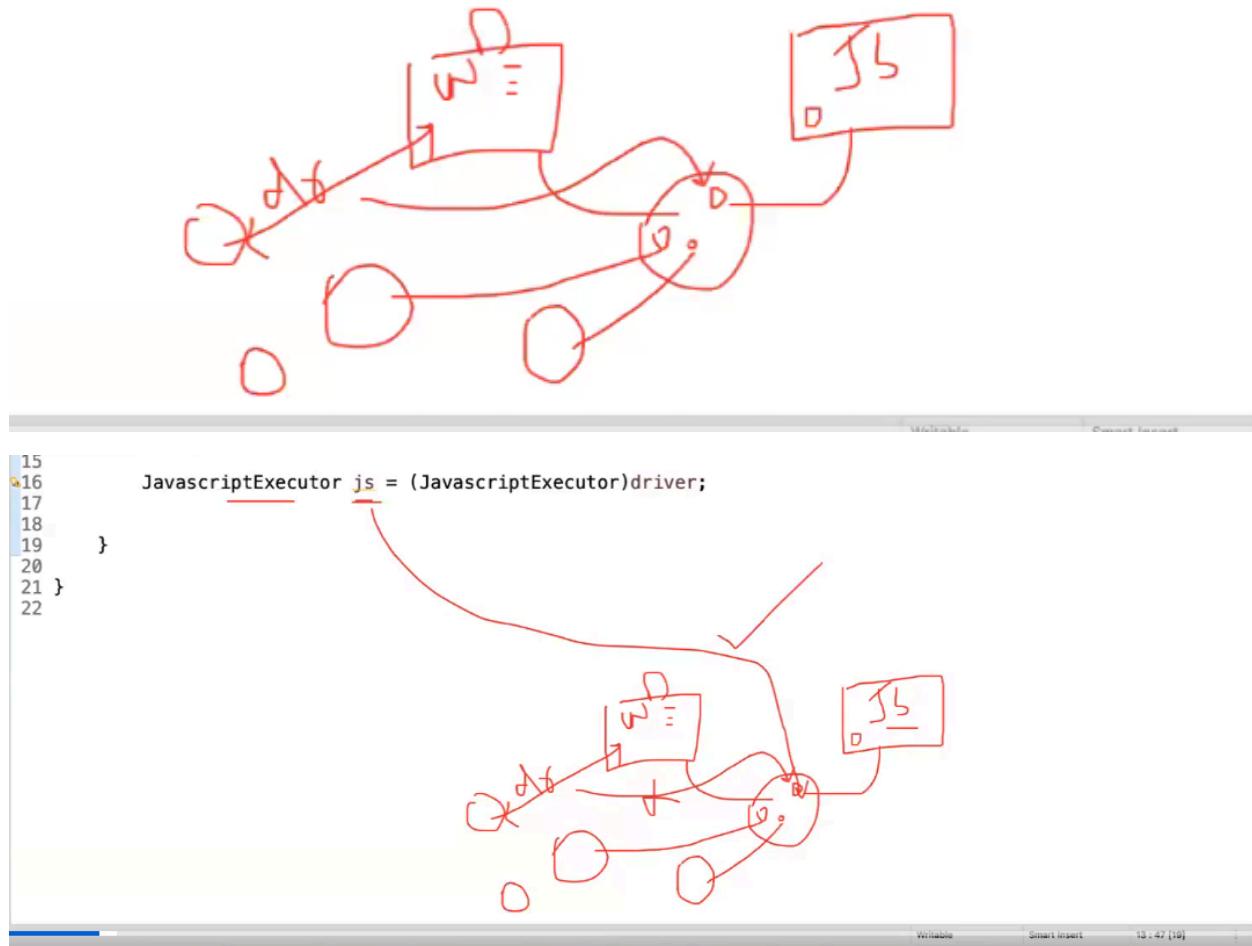
```

469 }
470
471 @Override
472 public Object executeScript(String script, Object... args) {
473     List<Object> convertedArgs =
474         Stream.of(args).map(new WebElementToJsonConverter()).collect(Collectors.toList());
475
476     return execute(DriverCommand.EXECUTE_SCRIPT(script, convertedArgs)).getValue();
477 }
478
479 @Override
480 public Object executeAsyncScript(String script, Object... args) {
481     List<Object> convertedArgs =
482         Stream.of(args).map(new WebElementToJsonConverter()).collect(Collectors.toList());
483
484     return execute(DriverCommand.EXECUTE_ASYNC_SCRIPT(script, convertedArgs)).getValue();
485 }
486 
```

Same like **has authenticator** method we need to convert **webdriver** interface to **js executor** interface.

Pictorial representation below-

Once converted the driver can access all methods of js executor plus webdriver.

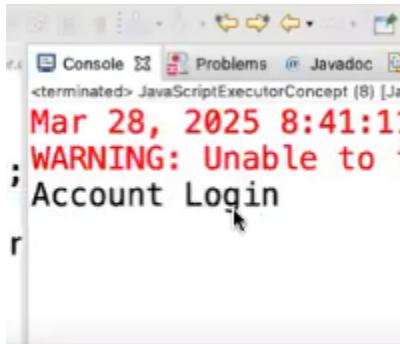


Naveen website.
Login page.
Get title.

The screenshot displays a web browser window titled "Account Login" with the URL "naveenautomationlabs.com/opencart/index.php?route=account/login". The page content includes the Naveen Automation Labs logo, navigation links for Desktops, Laptops & Notebooks, and Components, and a "New Customer" registration form. A developer's toolbar is open, showing the "Console" tab which displays the command "document.title" and its output, "Account Login". Below the browser is a code editor window showing a Java script named "JavaScriptExecutorConcept.java". The code imports necessary Selenium packages and uses the JavaScriptExecutor to inject JS code into the browser to retrieve the document title.

```
1 package seleniumsessions;
2
3 import org.openqa.selenium.JavascriptExecutor;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.chrome.ChromeDriver;
6
7 public class JavaScriptExecutorConcept {
8
9     public static void main(String[] args) {
10
11         //sel+java --> JavaScriptExecutor -- inject JS code into the browser
12
13         WebDriver driver = new ChromeDriver();
14         driver.get("https://naveenautomationlabs.com/opencart/index.php?route=account/login");
15
16         JavascriptExecutor js = (JavascriptExecutor)driver;
17         String title = js.executeScript("return document.title;").toString();
18         System.out.println(title);
19     }
20
21 }
22
```

Return document.title - is js return statement. We convert to string and store in variable and print it. This is converted so java can understand js code.



The screenshot shows the Eclipse IDE interface with the Java console tab selected. The output window displays the following text:

```
Mar 28, 2025 8:41:11  
WARNING: Unable to  
; Account Login
```

Get url of the page-

Run on console for same naveen login.

```
> document.URL  
<-- 'https://naveenautomationlabs.com/opencart/index.php?route=account/login'  
> |
```

Converted to code-

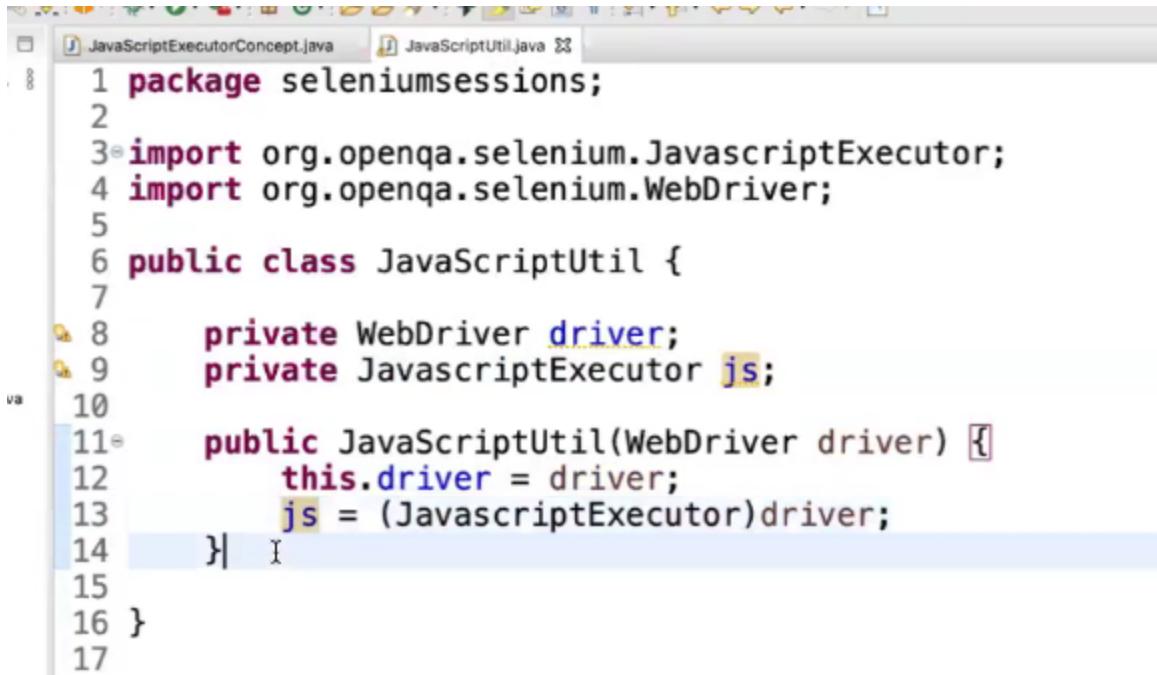


```
19  
20     String url = js.executeScript("return document.URL;").toString();  
21     System.out.println(url);  
22
```

The screenshot shows the Eclipse IDE interface with the Java console tab selected. The output window displays the following text:

```
Mar 28, 2025 8:42:13 AM org.openqa.selenium.devtools.CdpVersionFinder fin  
WARNING: Unable to find an exact match for CDP version 134, returning the  
; Account Login  
https://naveenautomationlabs.com/opencart/index.php?route=account/login
```

Create new class called js util for all js related utility code-



A screenshot of an IDE showing a Java file named `JavaScriptExecutorConcept.java`. The code defines a class `JavaScriptUtil` with a constructor that takes a `WebDriver` object and initializes it to `this.driver`. It also initializes a `JavascriptExecutor` object to `js`, which is cast from `this.driver`. The code is numbered from 1 to 17.

```
1 package seleniumsessions;
2
3 import org.openqa.selenium.JavascriptExecutor;
4 import org.openqa.selenium.WebDriver;
5
6 public class JavaScriptUtil {
7
8     private WebDriver driver;
9     private JavascriptExecutor js;
10
11    public JavaScriptUtil(WebDriver driver) {
12        this.driver = driver;
13        js = (JavascriptExecutor)driver;
14    }
15
16 }
17
```

Changed line 13 to `this.driver`, else we get warning but can be ignored. Both above and below line 13 is correct.

Created method to get the title.

```
10
11    public JavaScriptUtil(WebDriver driver) {
12        this.driver = driver;
13        js = (JavascriptExecutor)this.driver;
14    }
15
16    public String getTitleByJS() {
17        return js.executeScript("return document.title;").toString();
18    }
19
```

Url method-

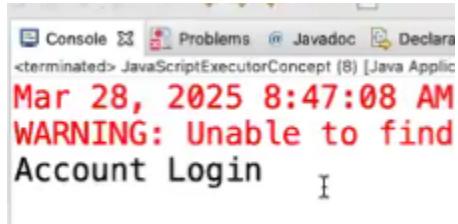
Create method to get url.

```
20
21
22    public String getURLByJS() {
23        return js.executeScript("return document.URL;").toString();
24    }
25
```

Calling title in test class-

Class name-JavaScriptExecutorConcept

```
12
13     WebDriver driver = new ChromeDriver();
14     driver.get("https://naveenautomationlabs.com/opencart/index.php?route=account/login");
15
16 //     JavascriptExecutor js = (JavascriptExecutor)driver;
17 //     String title = js.executeScript("return document.title;").toString();
18 //     System.out.println(title);
19 //
20 //     String url = js.executeScript("return document.URL;").toString();
21 //     System.out.println(url);
22
23
24     JavaScriptUtil jsUtil = new JavaScriptUtil(driver);
25     String title = jsUtil.getTitleByJS();
26     System.out.println(title);
27
```

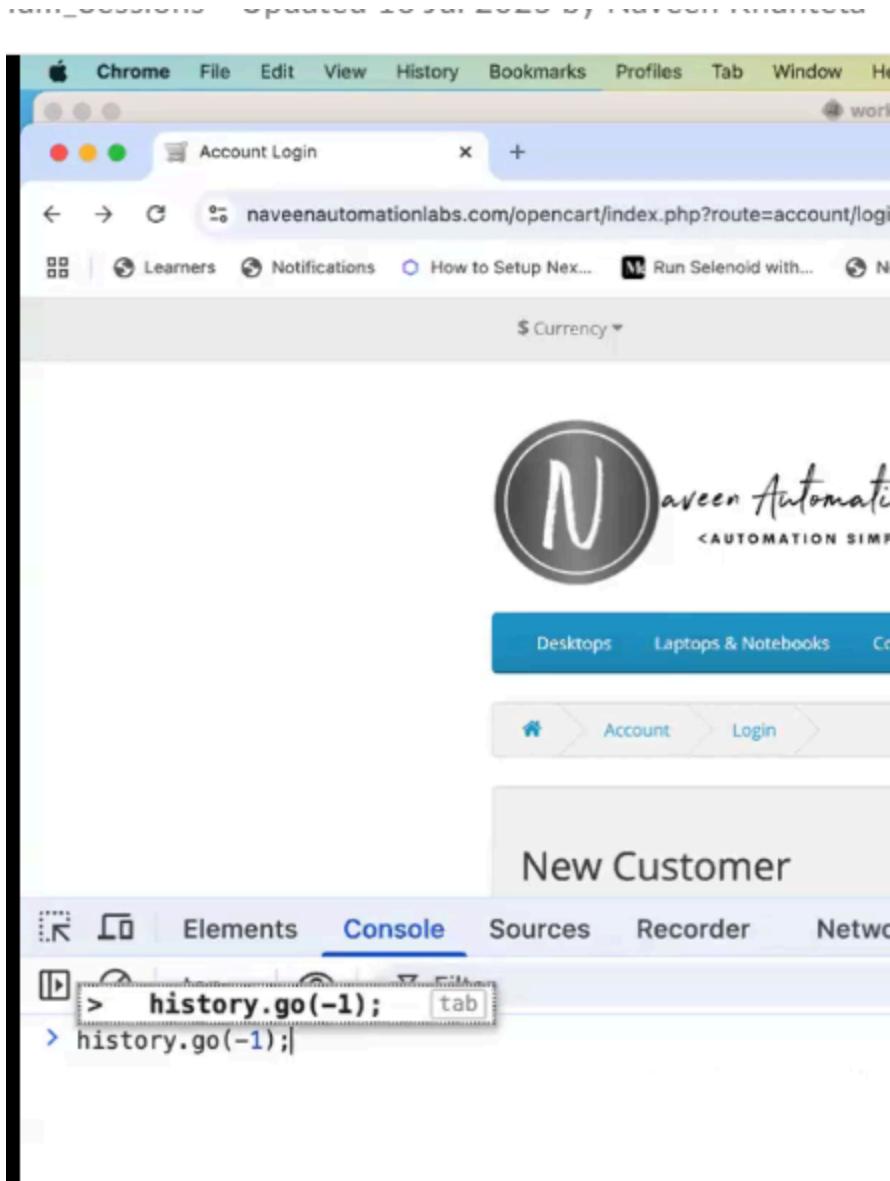


The screenshot shows a Java application running in an IDE. The status bar at the top indicates it's terminated. The main area displays a warning message in red text:

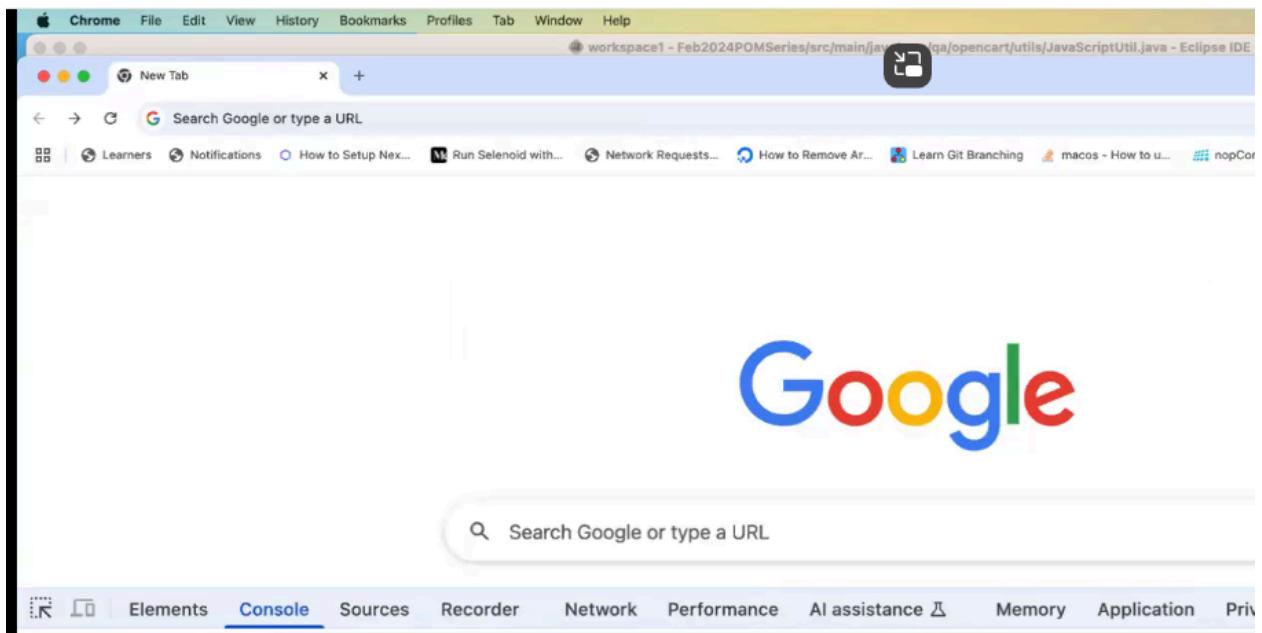
Mar 28, 2025 8:47:08 AM
WARNING: Unable to find
Account Login

History -1→

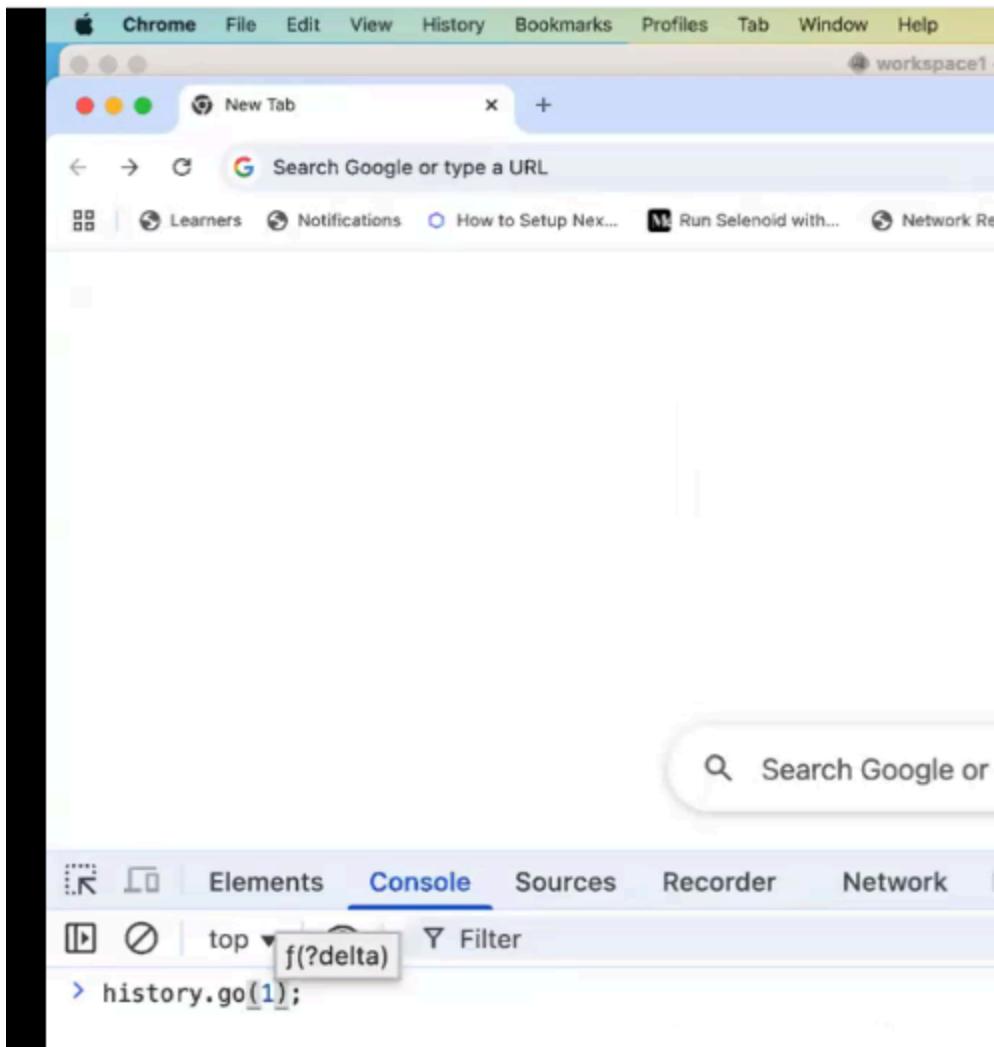
Go to naveen then google then naveen again.



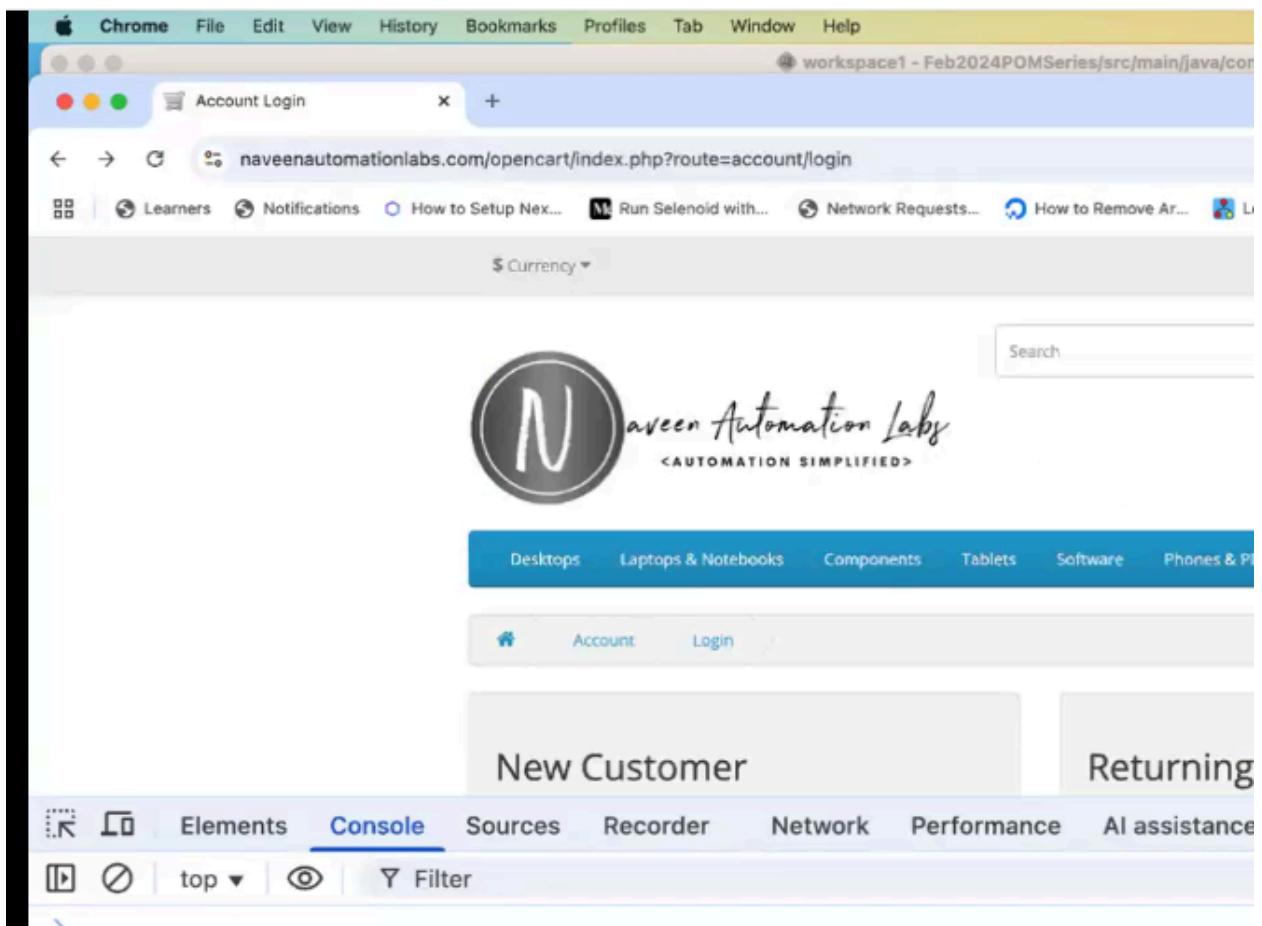
Click back button on browser and goes to earlier page.



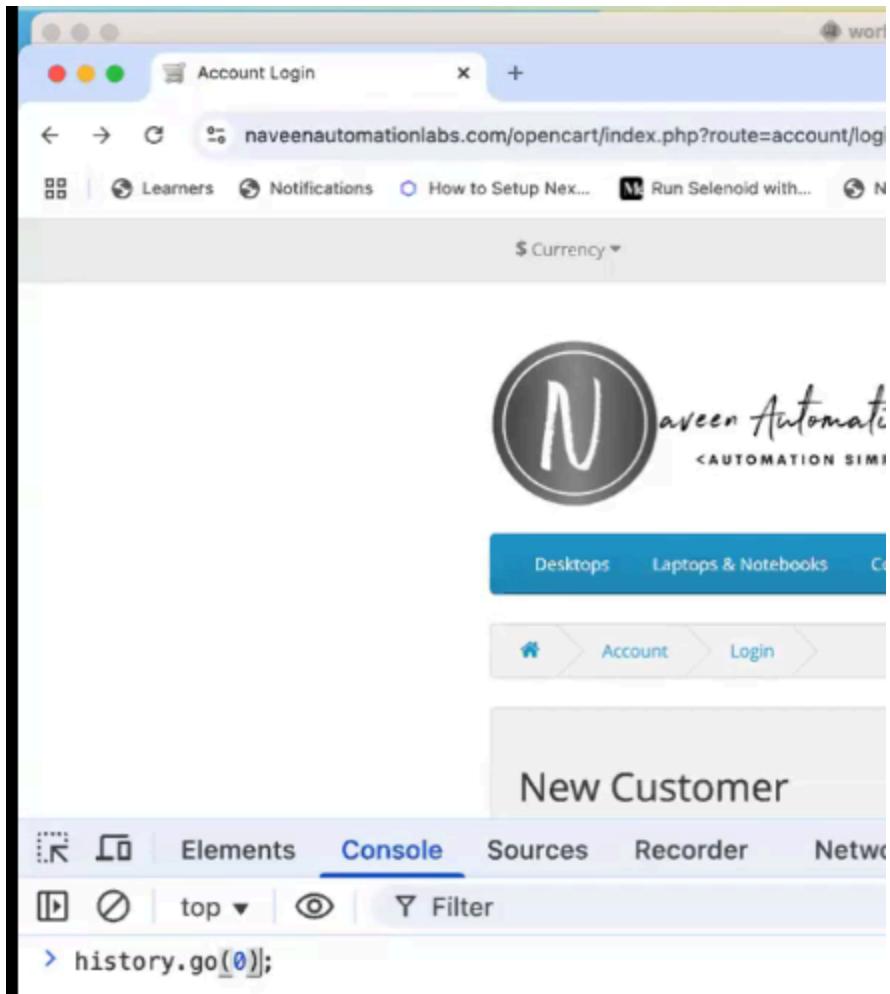
History 1→



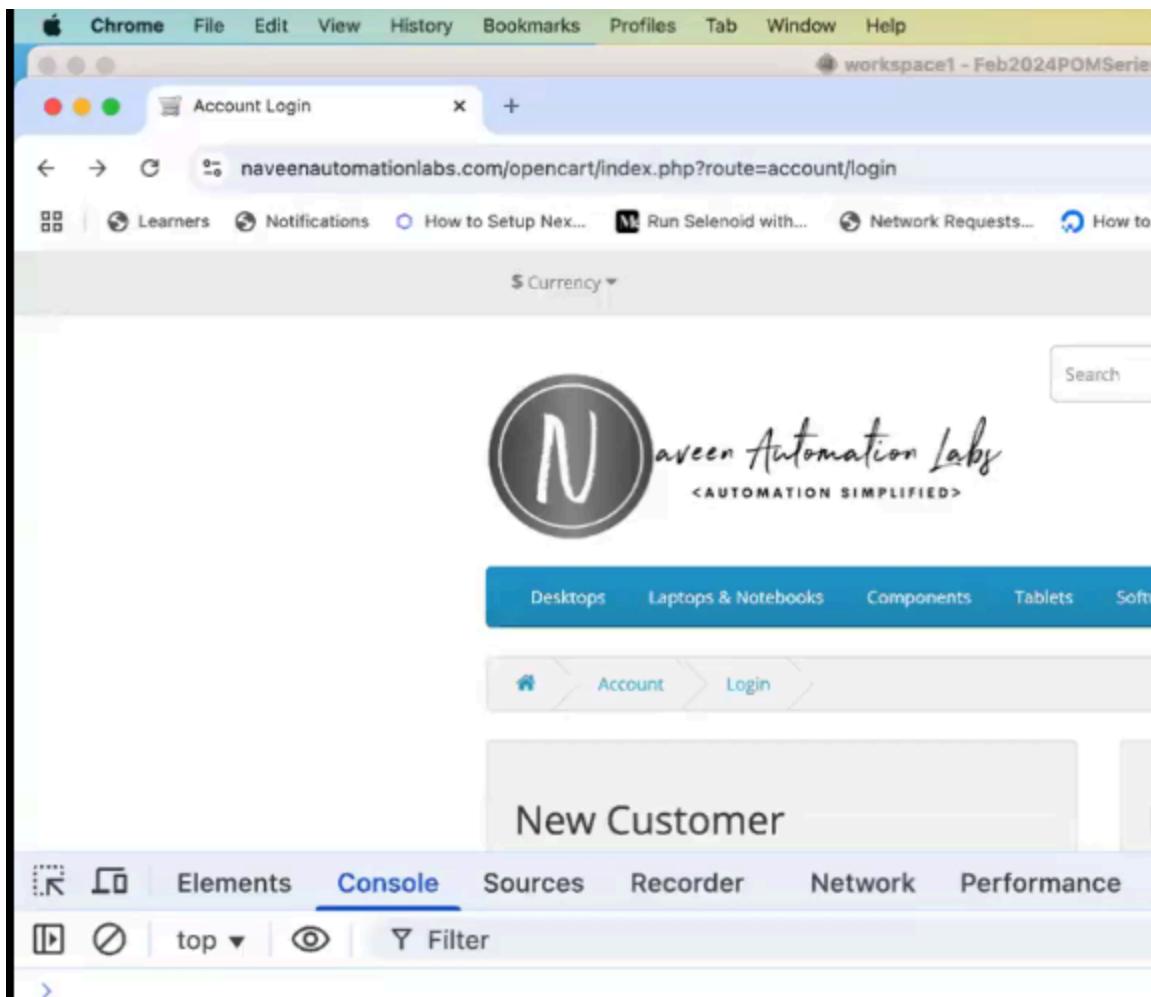
Clicks forward and goes to naveen website



Refresh current page-



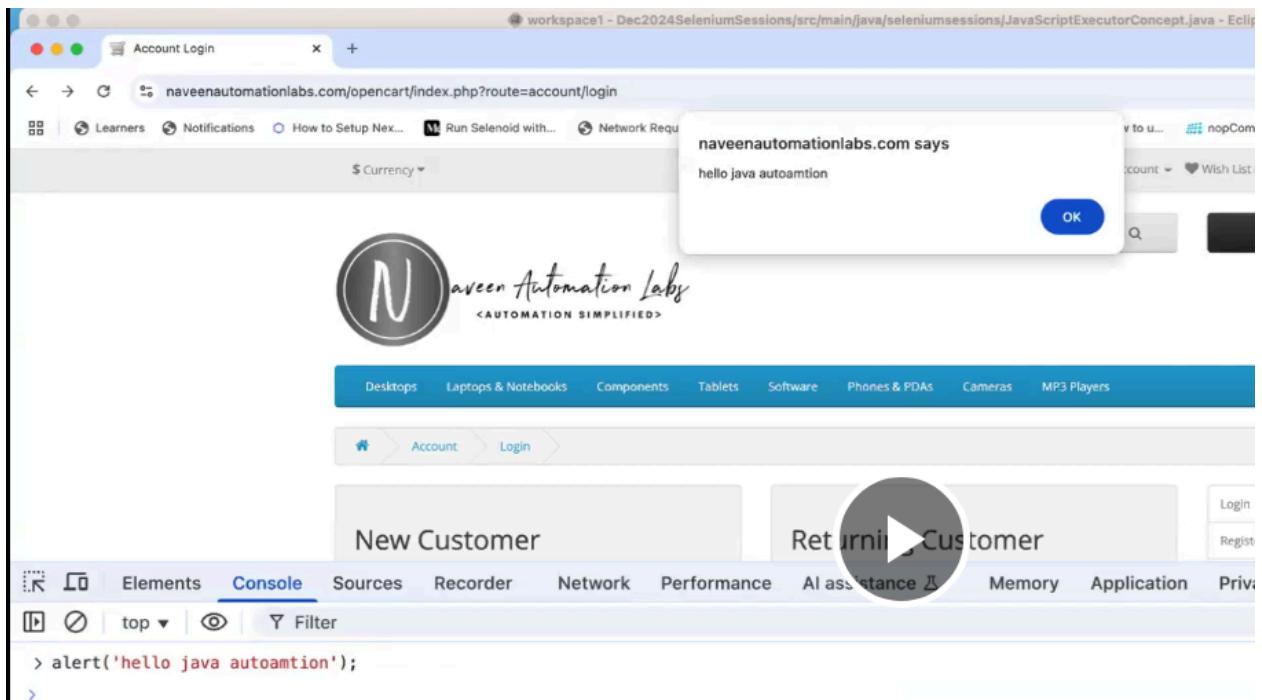
refreshed.



Add these methods in js util

```
25
24  public void refreshBrowserByJS() {
25      js.executeScript("history.go(0)");
26  }
27
28  public void navigateToBackPage() {
29      js.executeScript("history.go(-1)");
30  }
31
32  public void navigateToForwardPage() {
33      js.executeScript("history.go(1)");
34 }
```

Generate alert-



Add alert code in js util

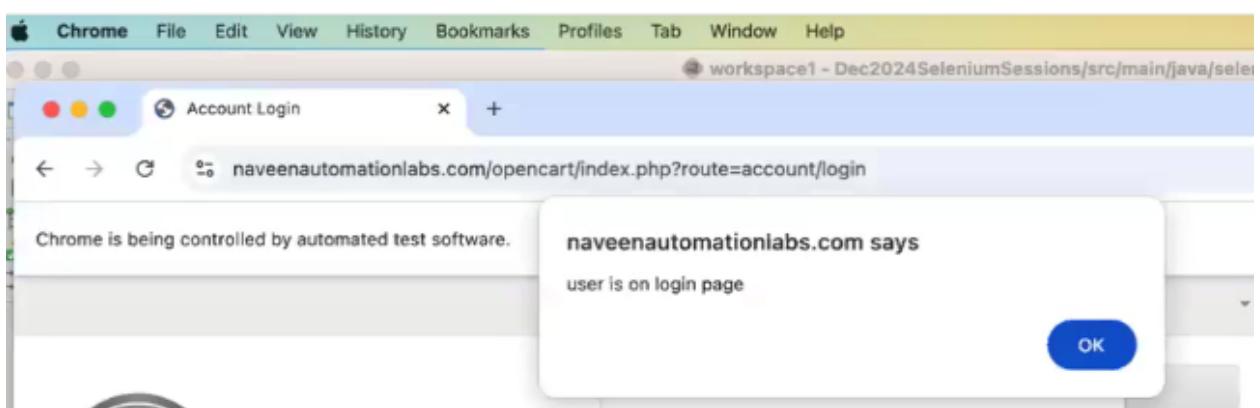
```
36  public void generateJSAlert(String message) {  
37      js.executeScript("alert('" + message + "')");  
38  }
```

Call in test class-

Class name -JavaScriptExecutorConcept

Naveen website login page.

```
27 |  
28     jsUtil.generateJSAlert("user is on login page");  
29 |
```



Update alert util method-

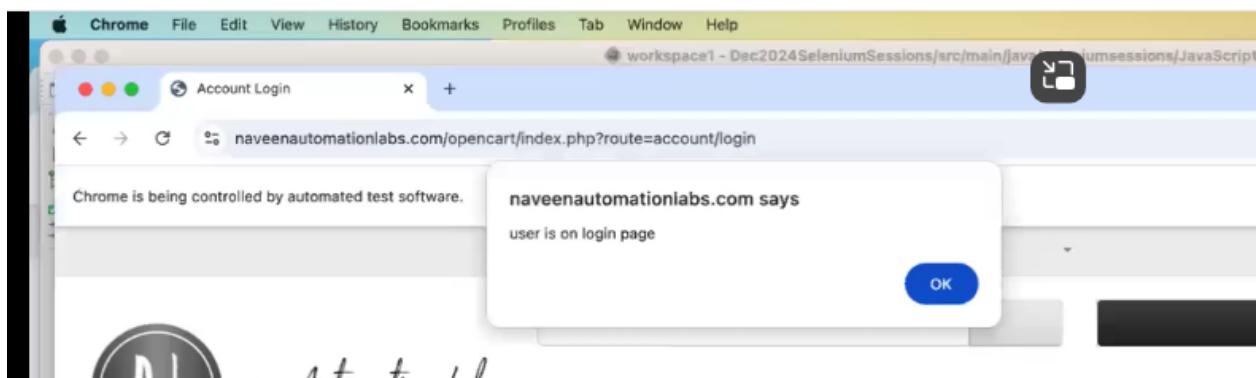
Added sleep to see the alert message, and then closed alert.

```
36  public void generateJSAlert(String message) {  
37      js.executeScript("alert('" + message + "')");  
38      try {  
39          Thread.sleep(2000);  
40      } catch (InterruptedException e) {  
41          e.printStackTrace();  
42      }  
43      driver.switchTo().alert().dismiss();  
44  }
```

This alert will now go off after 2 seconds when the alert.dismiss statement runs, else the alert will be there only and we cannot perform any further operation.

Run the same test code.

Class name -JavaScriptExecutorConcept



Gives all text of the webpage-

```

<!-- document.documentElement.innerText -->
<!-- $ Currency \n 123456789 My Account Wish List (0) Shopping Cart Checkout\n 0 item(s) - $0.00\nDesktops\nLaptops & Notebooks\nComponents\nTables\nSoftware\nPhones & PDAs\nCameras\nMP3 Players -->
<!-- Account Login -->
<!-- naveenautomationlabs.com/opencart/index.php?route=account/login -->
<!-- Run Selenium with... Network Requests... How to Remove Ar... Learn Git Branching macos - How to u... nopCommerce de... Maven Commands 1CRM: Projects All Bookmarks -->
<!-- Search -->
<!-- 123M00881 My Account Wish List Shopping Cart Checkout -->
<!-- New Customer -->
<!-- Register Account -->
<!-- By creating an account you will be able to shop faster, be up to date on an order's status, and keep track of the orders you have previously made. -->
<!-- Continue -->
<!-- Returning Customer -->
<!-- I am a returning customer -->
<!-- E-Mail Address -->
<!-- ne_vivekanand@gmail.com -->
<!-- Password -->
<!-- Log In -->
<!-- Login Register -->
<!-- Forgotten Password -->
<!-- My Account Address Book Wish List Order History Downloads Recurring payments Reward Points Returns Transactions Newsletter Information About Us Delivery Information Privacy Policy Terms & Conditions Customer Service Contact Us Returns Site Map Extras Brands Gift Certificates Affiliate Special My Account My Account Order History Wish List Newsletter Powered By OpenCart naveenopencart © 2025 -->

```

Another website-

```

<!-- document.documentElement.innerText -->
<!-- Home Sign Up \n Pricing \n Features \n Customers \n Contact \n Power up your sales, customer service, and marketing \n Add some magic to CRM \n Your business will soar with CRM \n Contact Management \n TRACK EVERYTHING INSTANTLY \n Keep a single shared customer profile, company-wide. \n Deal Pipeline \n SALES PIPELINE DASHBOARD \n Get your finger on the pulse with graphs and charts detailing sales and forecasts. \n Marketing Automation \n UNLIMITED CAMPAIGNS \n Track conversions end to end with personalized email, direct mail, voice, and print using a personalized touch. \n Get Mobile \n ANY DEVICE \n Works on any device – just fire up your mobile browser, no app required. Mobile bookmark Free CRM for quick access. \n Companies & Contacts \n Our deal tracking tool and sales pipeline dashboard give you a current, comprehensive view of all your sales activities. Know where every customer is in the sales cycle, deal size, contact history, even competitor information to help craft more effective CRM messaging. Reports & Dashboards provide a smooth, accurate read of everything happening. \n Deals & Sales Pipeline \n Our deal tracking tool and sales pipeline dashboard give you a current, comprehensive view of all your sales activities. Know where every customer is in the sales cycle, deal size, contact history, even competitor information to help craft more effective CRM messaging. Reports & Dashboards provide a smooth, accurate read of everything happening. \n Calls & Voicemail \n Free CRM has powerful call tracking and call scheduling features that make it easy to manage your calls. You can also use our call reporting feature to follow up, scheduling and call reporting with detailed call center features. \n Forms \n Forms are questionnaires and answer surveys and can also be call scripts where data can be explained and inputted by your sales or support staff. You can also distribute the forms to clients through a URL to a webpage to get customer feedback. Form and survey creation is natural with our form builder, creating "Wizard Forms" – multi-page forms that are easy to follow and complete. \n Reports \n Reporting features allow you to gauge the effectiveness of your software solution and how well your staff is completing the sales cycle. You can also run reports that track system use for each of your team members, allowing you to see where additional training is needed or extra customization in the application so that your team can use the tool more efficiently and smoothly as it pertains to your normal business operation. \n Marketing Campaigns \n You can develop targeted marketing campaigns based on customer profile and history. Execute email campaigns around time-sensitive promotions and opportunities with unlimited email. Track marketing in real-time, focus and improve the quality of your lead generation. Print, SMS, calling and email campaigns are all included. Utilize our Promotions feature to track loyalty and give customers benefits. \n Task & Event Tracking \n Tracking tasks, issues, and sales give you complete visibility into the -->
```

Add in js util-

```

46+     public String getPageInnerText() {
47+         return js.executeScript("return document.documentElement.innerText;").toString();
48     }

```

Test class-

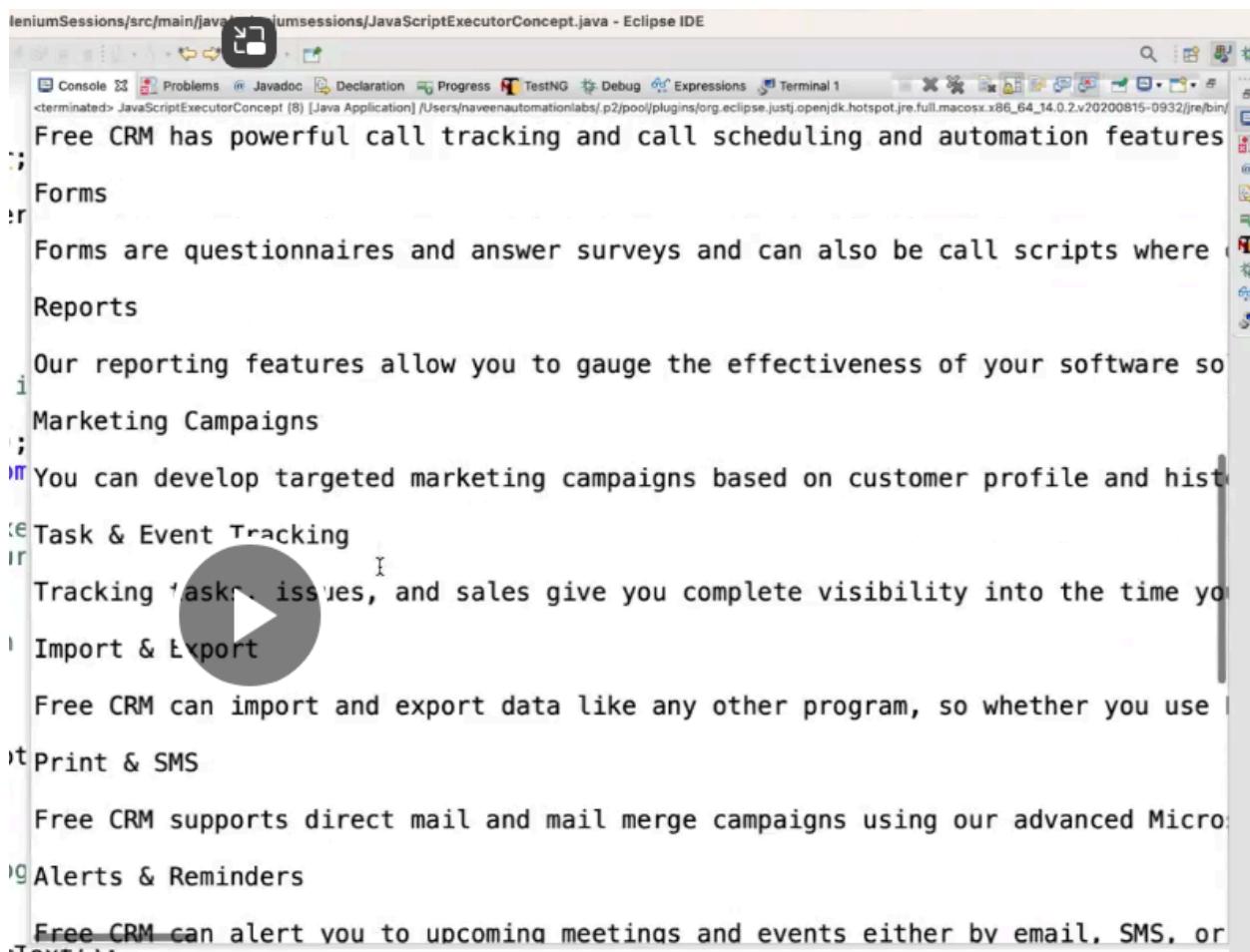
Class name -JavaScriptExecutorConcept

Change url.

```
12
13     WebDriver driver = new ChromeDriver();
14     driver.get("https://classic.crmpro.com/");
15
```

Call inner text-

```
23
24     JavaScriptUtil jsUtil = new JavaScriptUtil(driver);
25 //     String title = jsUtil.getTitleByJS();
26 //     System.out.println(title);
27 //
28 //     jsUtil.generateJSAlert("user is on login page");
29
30
31     String pageText = jsUtil.getPageInnerText();
32     System.out.println(pageText);
```

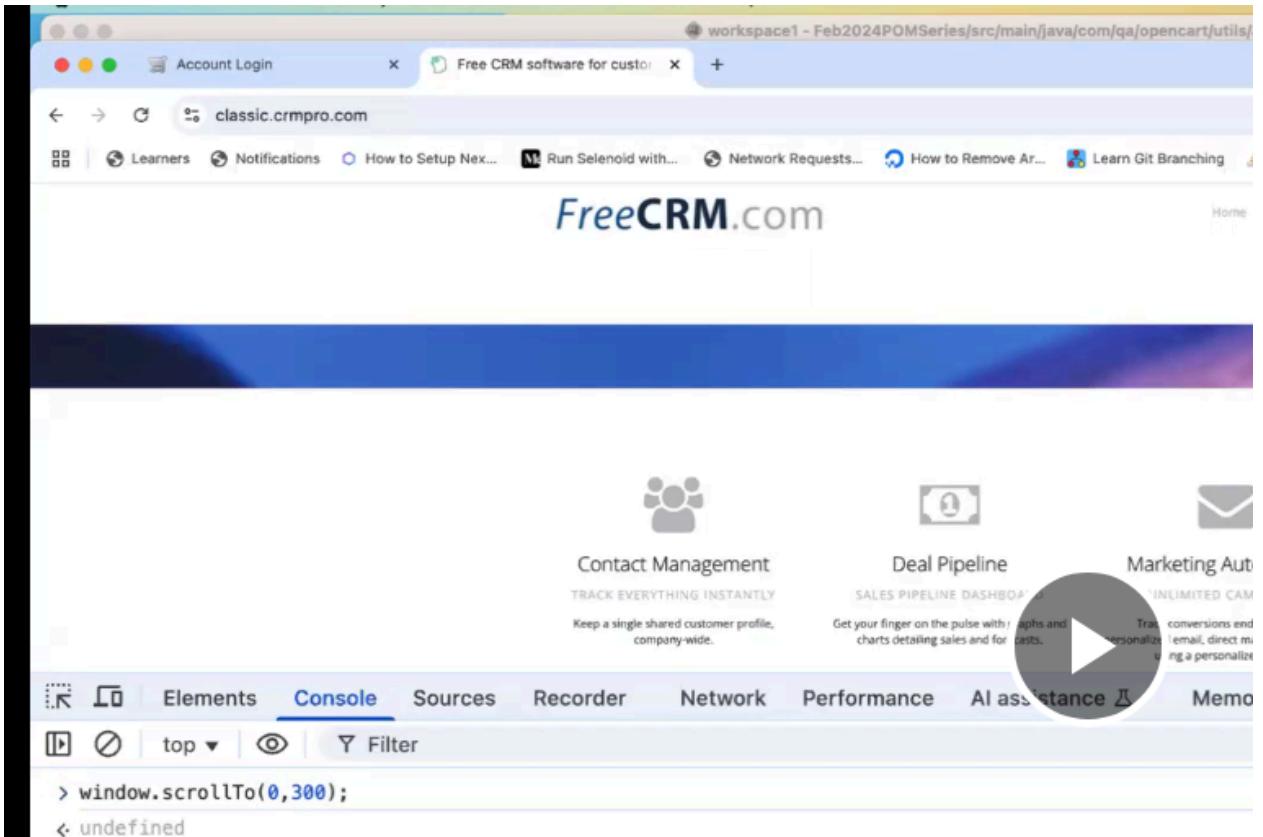


The screenshot shows the Eclipse IDE interface with a Java code editor and a terminal window. The code is as follows:

```
leniumSessions/src/main/java/com/naveenautomationlabs/JavaScriptExecutorConcept.java - Eclipse IDE
Console Problems Javadoc Declaration Progress TestNG Debug Expressions Terminal 1
terminated: JavaScriptExecutorConcept [8] [Java Application] /Users/naveenautomationlabs/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_14.0.2.v20200815-0932/re/bin/
Free CRM has powerful call tracking and call scheduling and automation features
;
Forms
Forms are questionnaires and answer surveys and can also be call scripts where
Reports
Our reporting features allow you to gauge the effectiveness of your software so
Marketing Campaigns
;
You can develop targeted marketing campaigns based on customer profile and hist
Task & Event Tracking
Tracking tasks, issues, and sales give you complete visibility into the time you
Import & Export
Free CRM can import and export data like any other program, so whether you use
Print & SMS
Free CRM supports direct mail and mail merge campaigns using our advanced Micro
Alerts & Reminders
Free CRM can alert you to upcoming meetings and events either by email, SMS, or
```

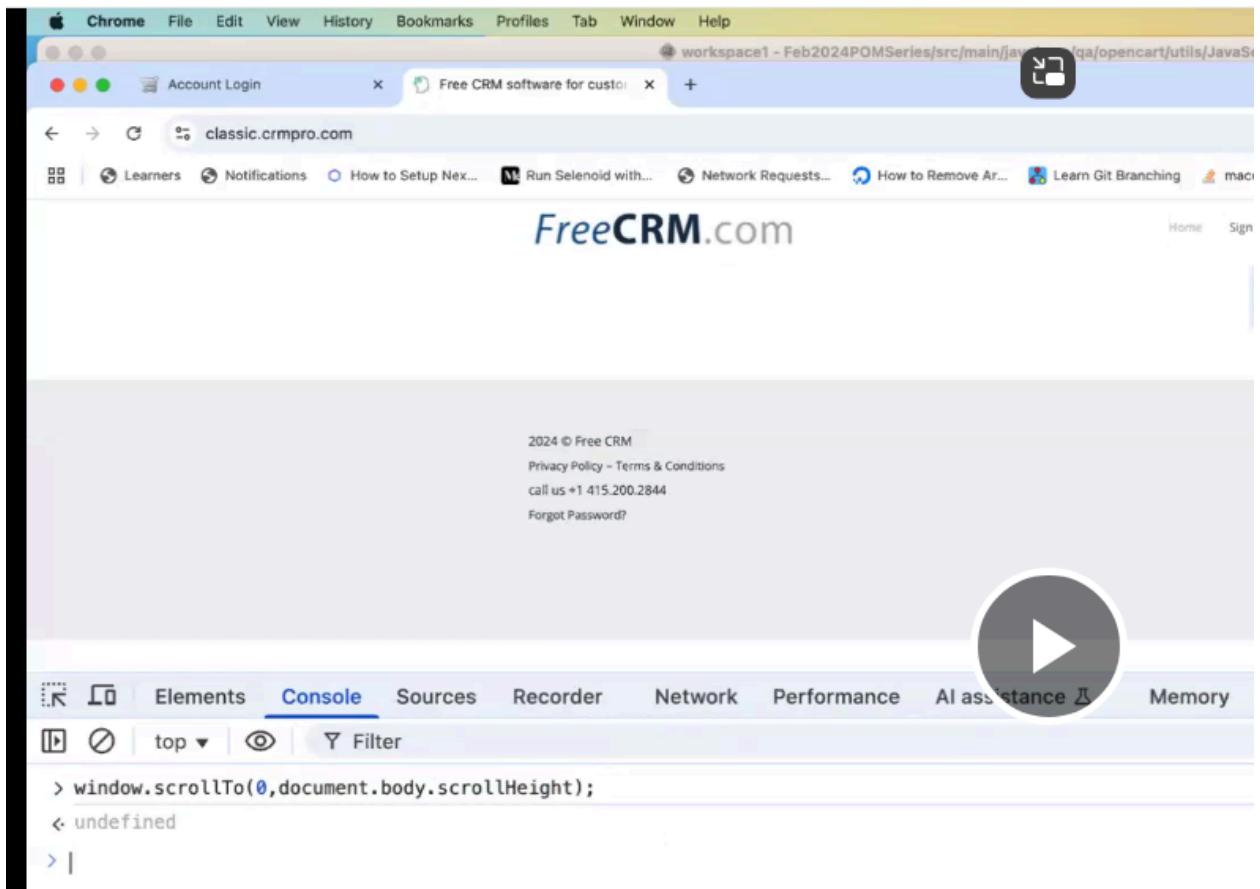
The terminal window shows the output of the Java application, which is the content of the HTML page from the URL specified in the code. A play button icon is overlaid on the terminal window.

Scroll down 300 pixels-



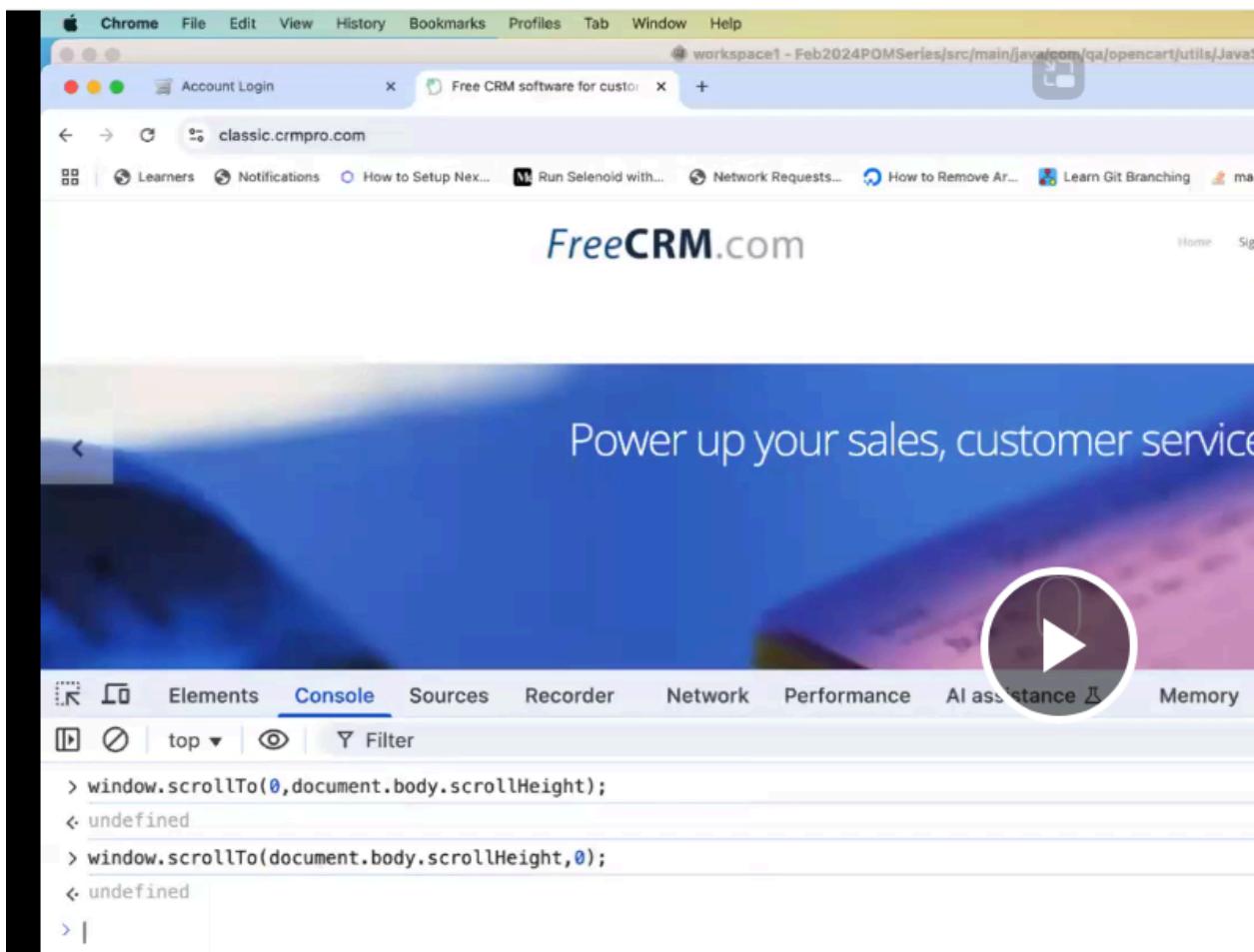
Scroll to bottom of page-

Start from zero and scroll to bottom.

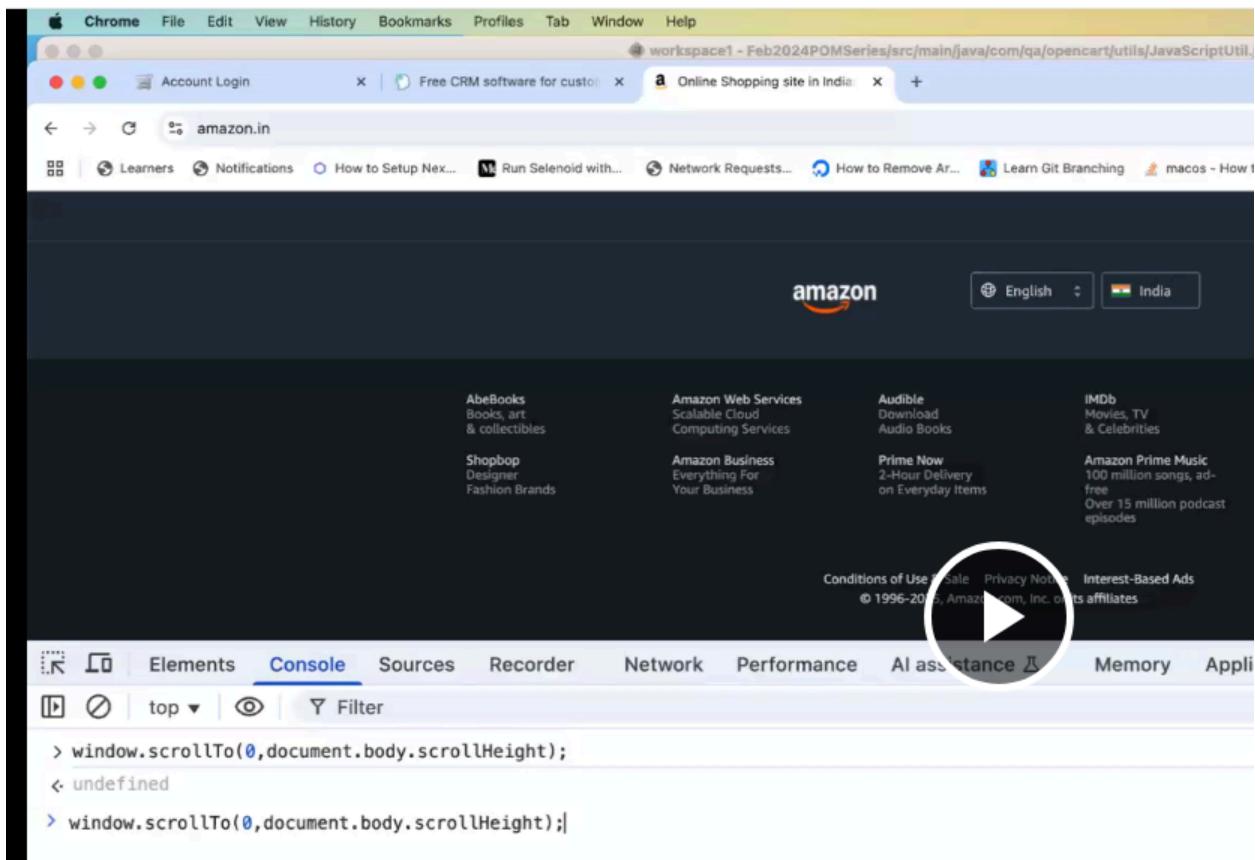


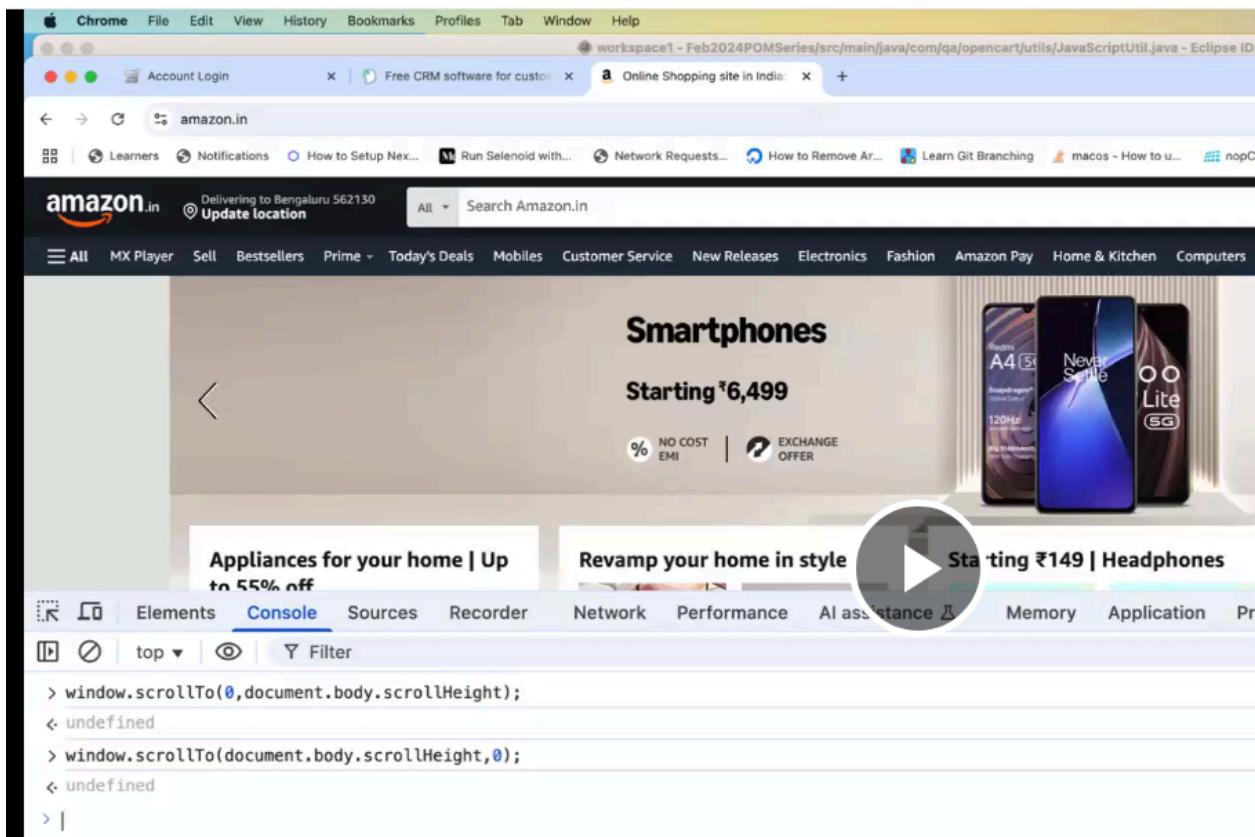
Bottom to top-

Reverse the x and y coordinates.



Works with amazon also-





Add to js util-

```

50  public void scrollPageDown() {
51      JavascriptExecutor js = (JavascriptExecutor) driver;
52      js.executeScript("window.scrollTo(0, document.body.scrollHeight)");
53  }
54
55  public void scrollPageDown(String height) {
56      JavascriptExecutor js = (JavascriptExecutor) driver;
57      js.executeScript("window.scrollTo(0, '" + height + "')");
58  }
59
60  public void scrollPageUp() {
61      JavascriptExecutor js = (JavascriptExecutor) driver;
62      js.executeScript("window.scrollTo(document.body.scrollHeight, 0)");
63  }
64

```

Test class-

Class name-JavaScriptExecutorConcept

```

12
13     WebDriver driver = new ChromeDriver();
14     driver.get("https://amazon.in/");
15

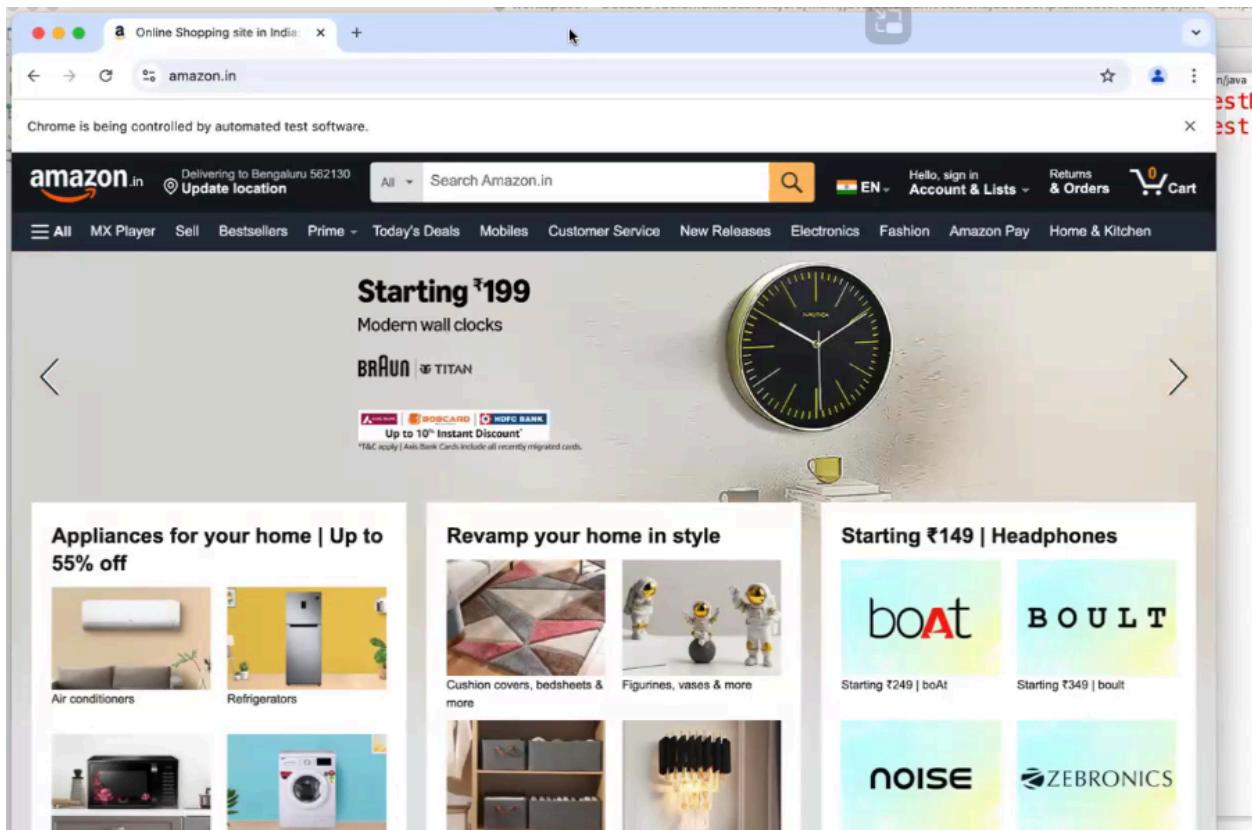
```

```

23
24     JavaScriptUtil jsUtil = new JavaScriptUtil(driver);
25     // 
26     }
27
28     jsUtil.scrollPageDown();
29     Thread.sleep(2000);
30     jsUtil.scrollPageUp();
31
32
33
34
35
36
37
38
39
40
41
42

```

It will scroll down and up.



Scroll into view- 40.00

Js util page.

Weird syntax, take the zeroth element and then scroll into view. If we pass false then it wont scroll to that element.

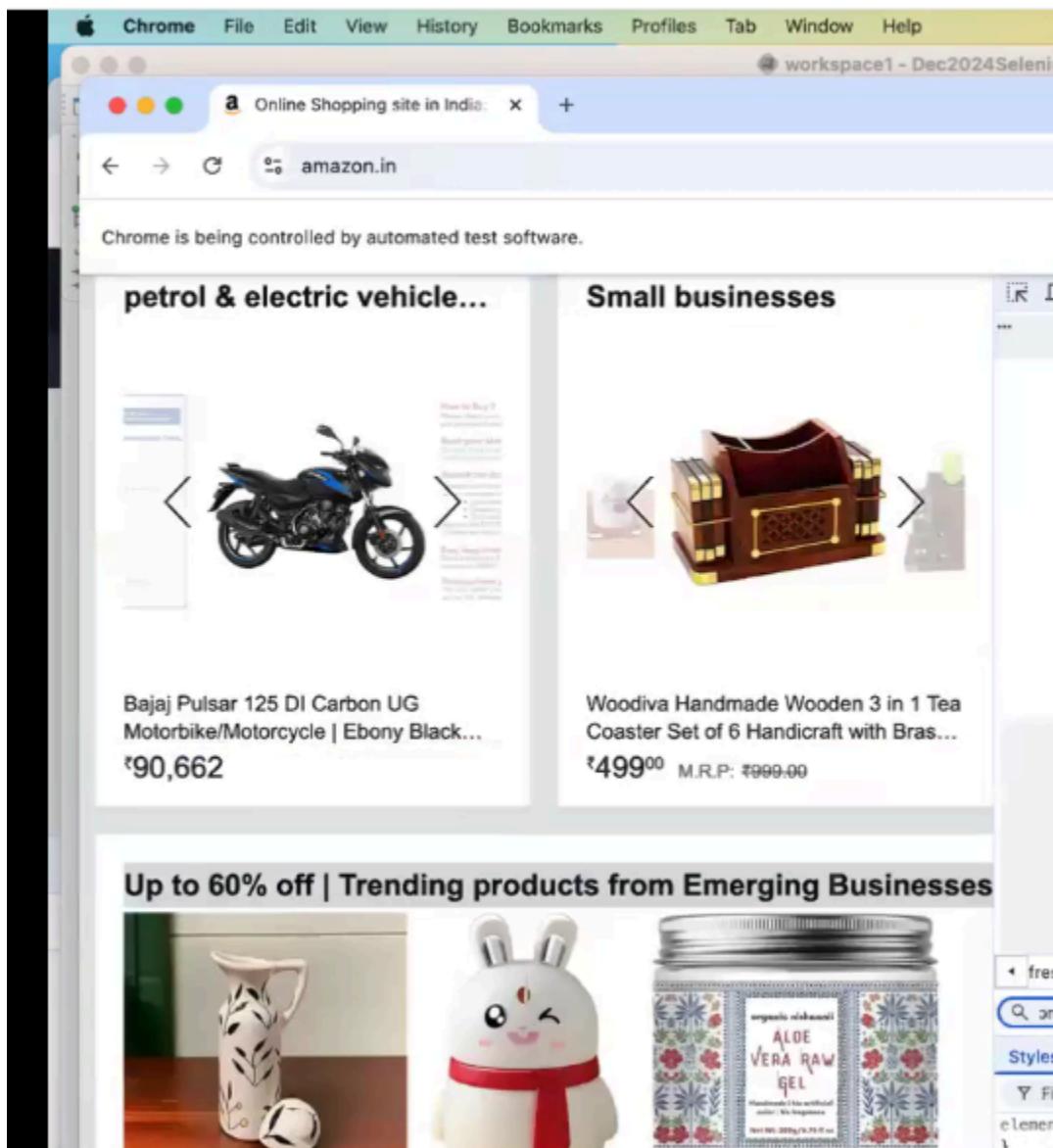
```
65  
66  public void scrollIntoView(WebElement element) {  
67      js.executeScript("arguments[0].scrollIntoView(true);", element);  
68  }
```

Test class-

Class name-JavaScriptExecutorConcept

Same amazon website.

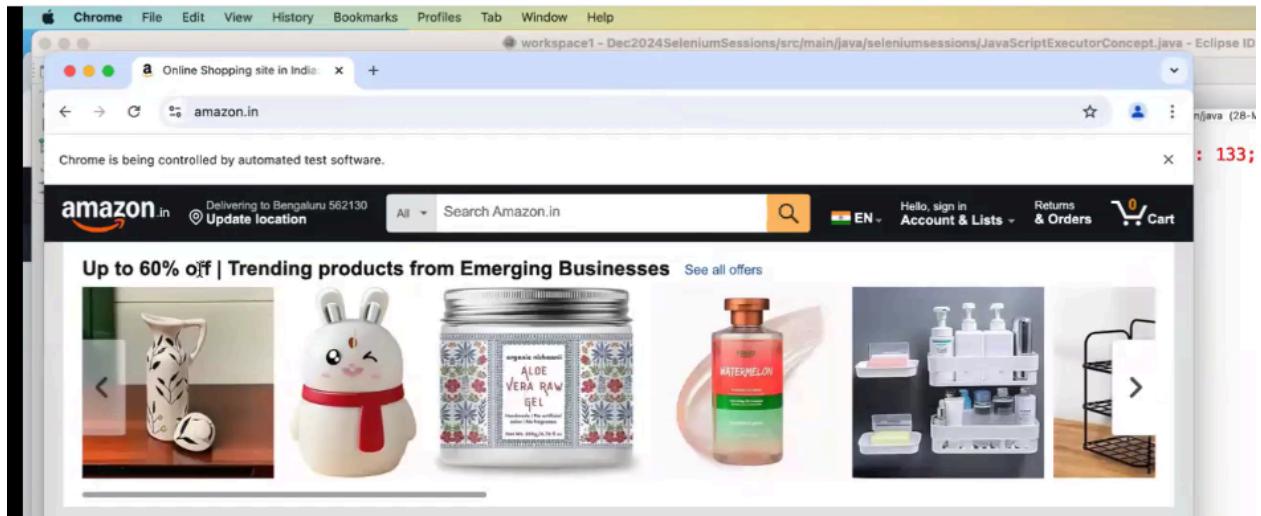
Amazon scroll. Upto certain element. Lets see upto the teddy bar container.



Test class-

Class name-JavaScriptExecutorConcept

```
24
25     JavaScriptUtil jsUtil = new JavaScriptUtil(driver);
26     String title = driver.getTitle();
27
28     WebElement element = driver.findElement(By.xpath("//span[contains(text(),'Trending products from Emerging Businesses')]"));
29
30     jsUtil.scrollIntoView(element);
31
32 }
```



Semicolon in js is not mandatory to write.

We can remove and it will work.

Js util-

Draw border.

Zeroth element, take the style and create.

```
69
70     public void drawBorder(WebElement element) {
71         js.executeScript("arguments[0].style.border='3px solid red'", element);
72     }
72 }
```

Test class-

Class name-JavaScriptExecutorConcept

```
13
14     WebDriver driver = new ChromeDriver();
15     driver.get("https://naveenautomationlabs.com/opencart/index.php?route=account/login");
16 }
```

```
24
```

```
25     JavaScriptUtil jsUtil = new JavaScriptUtil(driver);  
26     String title = driver.getTitle();
```

Border for email.

Naveen website.

48 - ranjani name.

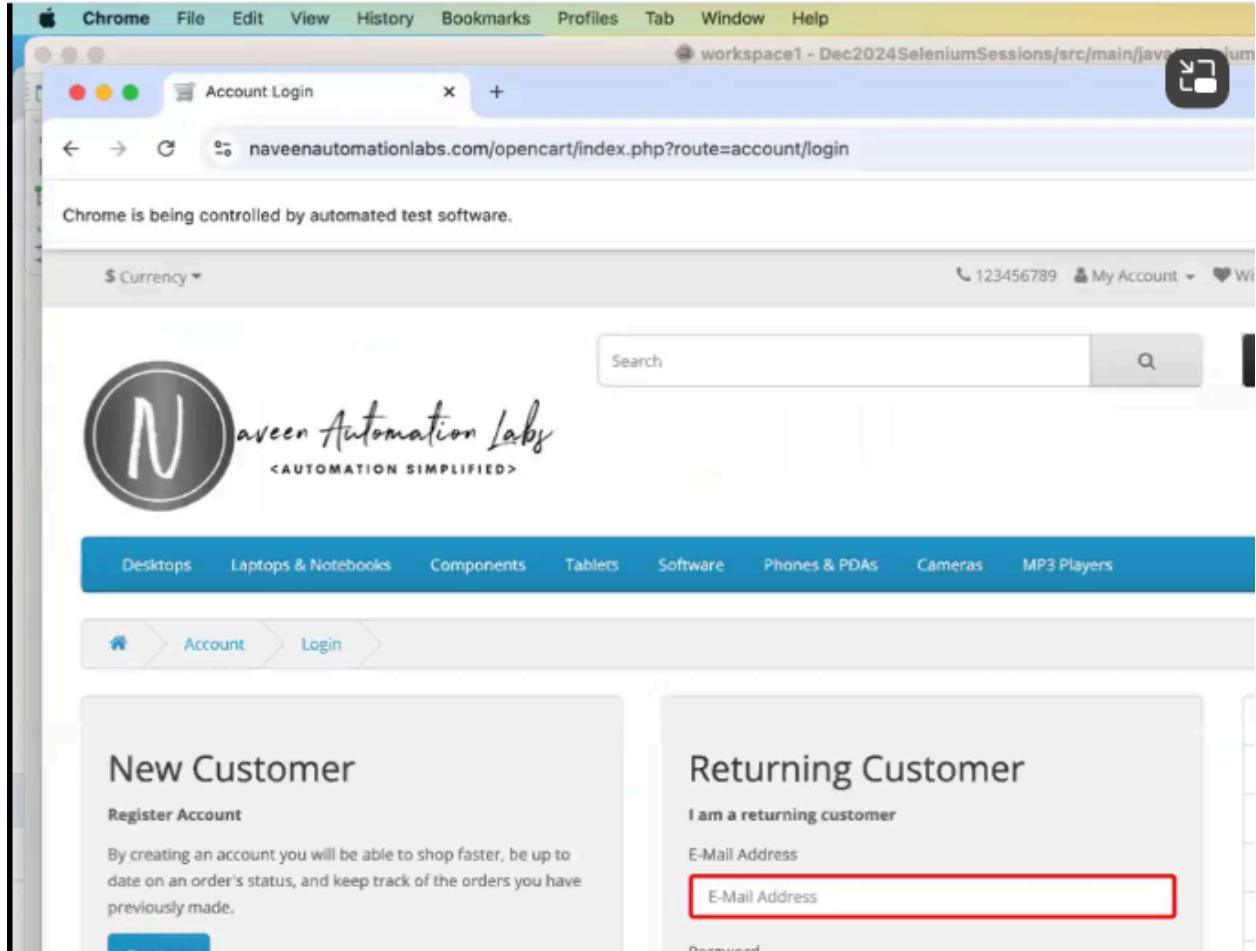
```
46
```

```
47     WebElement emailId = driver.findElement(By.id("input-email"));
```

```
48     jsUtil.drawBorder(emailId);
```

```
49
```

```
50
```

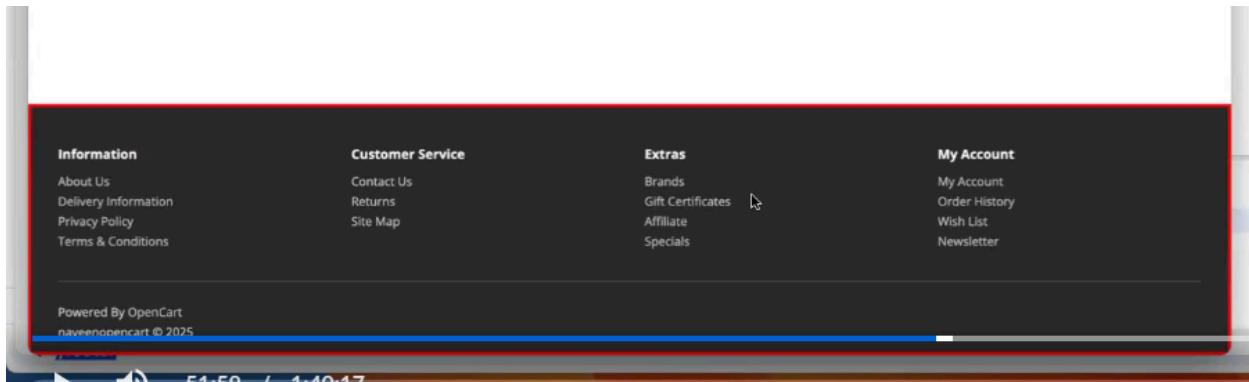


Create border for entire returning customer in naveen website-

Create border for footer also. Naveen website. Class name-JavaScriptExecutorConcept

```
46      WebElement emailId = driver.findElement(By.id("input-email"));
47      WebElement loginForm = driver.findElement(By.cssSelector("div.well form"));
48      WebElement footer = driver.findElement(By.xpath("//footer"));
49
50      jsUtil.drawBorder(emailId);
51      jsUtil.drawBorder(loginForm);
52      jsUtil.drawBorder(footer);
53 
```

The screenshot shows a Chrome browser window with the title "Account Login". The URL in the address bar is "naveenautomationlabs.com/opencart/index.php?route=account/login". A status message at the top says "Chrome is being controlled by automated test software." The page itself is the account login page for Naveen Automation Labs. It includes a logo, a search bar, and a navigation menu with links like Desktops, Laptops & Notebooks, Components, Tablets, Software, Phones & PDAs, Cameras, and MP3 Players. Below the menu is a breadcrumb navigation: Home > Account > Login. There are two main sections: "New Customer" on the left and "Returning Customer" on the right. The "Returning Customer" section contains fields for "E-Mail Address", "Password", and a "Login" button, all of which are highlighted with a red border.



Flash method-

Naveen login website.

Get background color of element.

Getcssvalue is the method name. Give the property name as parameter and returns string.

These stylings are seen in styles-

Property	Value
display	block
width	100%
height	34px
padding	6px 12px
font-size	14px
line-height	1.42857143
color	#555555
background-color	#cccccc
background-image	none
border	1px solid #cccccc
border-radius	4px
webkit-box-shadow	inset 0 1px 1px rgba(0, 0, 0, .075)
box-shadow	inset 0 1px 1px rgba(0, 0, 0, .075)

Call change color method. This is our own method.
Execute the script and pass the color to the element.
Wait for 20 ms for one color. Keep changing the
background color as we provide them from flash
method, this is like a disco.

No need to byheart these scripts. 53.00

Jsutil-

```
15
74= public void flash(WebElement element) {
75     String bgcolor = element.getCssValue("backgroundColor");//blue
76     for (int i = 0; i < 100; i++) {
77         changeColor("rgb(0,200,0)", element);// green
78         changeColor(bgcolor, element);// blue
79     }
80 }
81
82= private void changeColor(String color, WebElement element) {
83     JavascriptExecutor js = ((JavascriptExecutor) driver);
84     js.executeScript("arguments[0].style.backgroundColor = '" + color + "'", element);//GBGBGBGBGBGB
85
86     try {
87         Thread.sleep(20);
88     } catch (InterruptedException e) {
89     }
90 }
91
```

Naveen website.

Email field.

Password field.

Test class calling it-

Class name-JavaScriptExecutorConcept

```

46
47     WebElement emailId = driver.findElement(By.id("input-email"));
48     WebElement password = driver.findElement(By.id("input-password"));
49
50     WebElement loginForm = driver.findElement(By.cssSelector("div.well form"));
51     WebElement footer = driver.findElement(By.xpath("//footer"));
52
53 //      jsUtil.drawBorder(emailId);
54 //      jsUtil.drawBorder(loginForm);
55 //      jsUtil.drawBorder(footer);
56
57     jsUtil.flash(emailId);
58     emailId.sendKeys("test@gmail.com");
59
60     jsUtil.flash(password);
61     password.sendKeys("test@123");
62

```

Account Login

naveenautomationlabs.com/opencart/index.php?route=account/login

Chrome is being controlled by automated test software.

\$ Currency ▾ 123456789 My Account ▾

Search

Naveen Automation Labs
AUTOMATION SIMPLIFIED

Desktops Laptops & Notebooks Components Tablets Software Phones & PDAs Cameras MP3 Players

Account Login

New Customer

Register Account

By creating an account you will be able to shop faster, be up to date on an order's status, and keep track of the orders you have previously made.

Continue

Returning Customer

I am a returning customer

E-Mail Address

test@gmail.com

Password

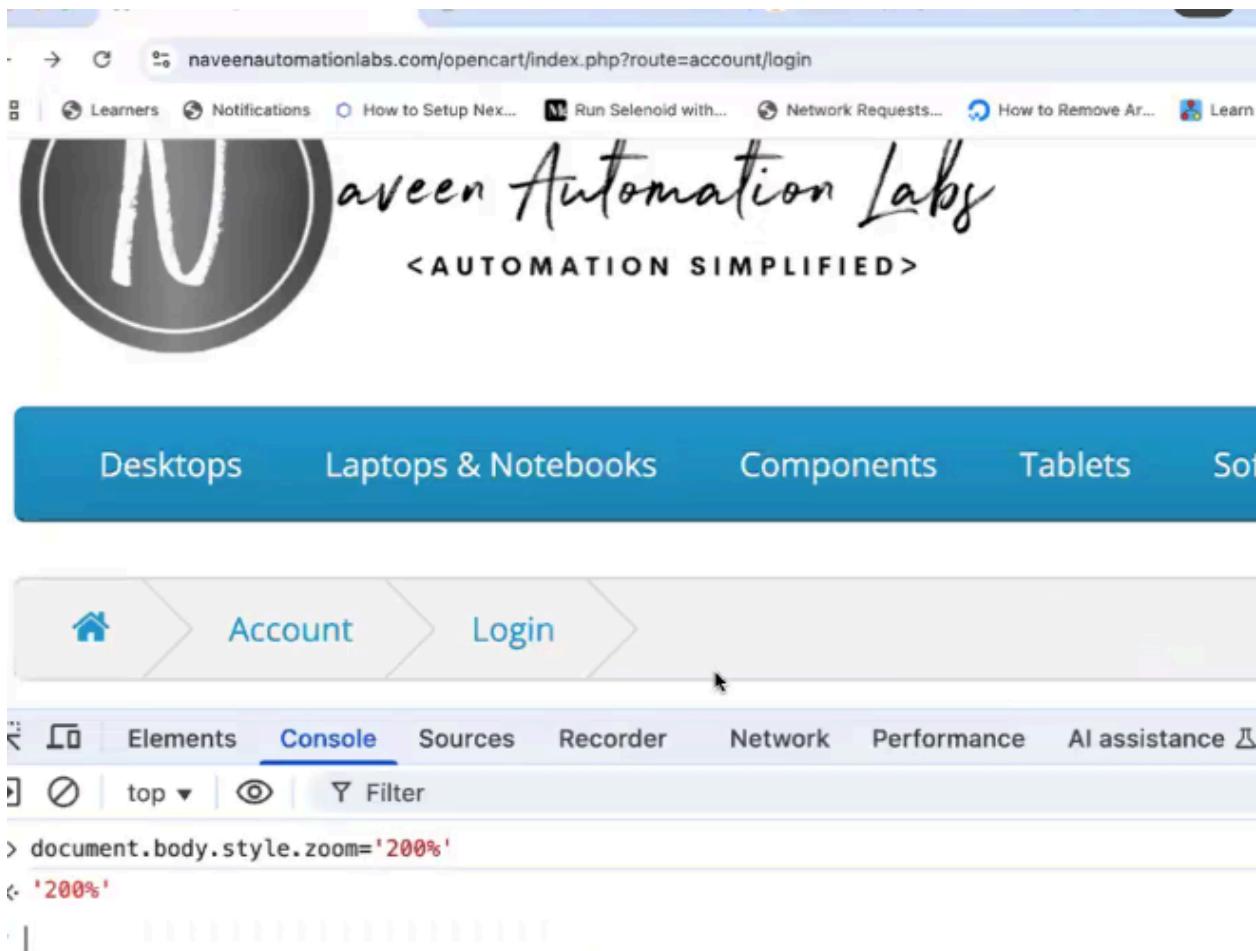
Forgotten Password

Login

Just blink for some time to see- 1.01
Reduce the size to 7.
Js util class-

```
73
74  public void flash(WebElement element) {
75      String bgcolor = element.getCssValue("backgroundColor");//blue
76      for (int i = 0; i < 7; i++) {
77          changeColor("rgb(0,200,0)", element);// green
78          changeColor(bgcolor, element);// blue
79      }
80  }
```

Zoom to 200 percent-

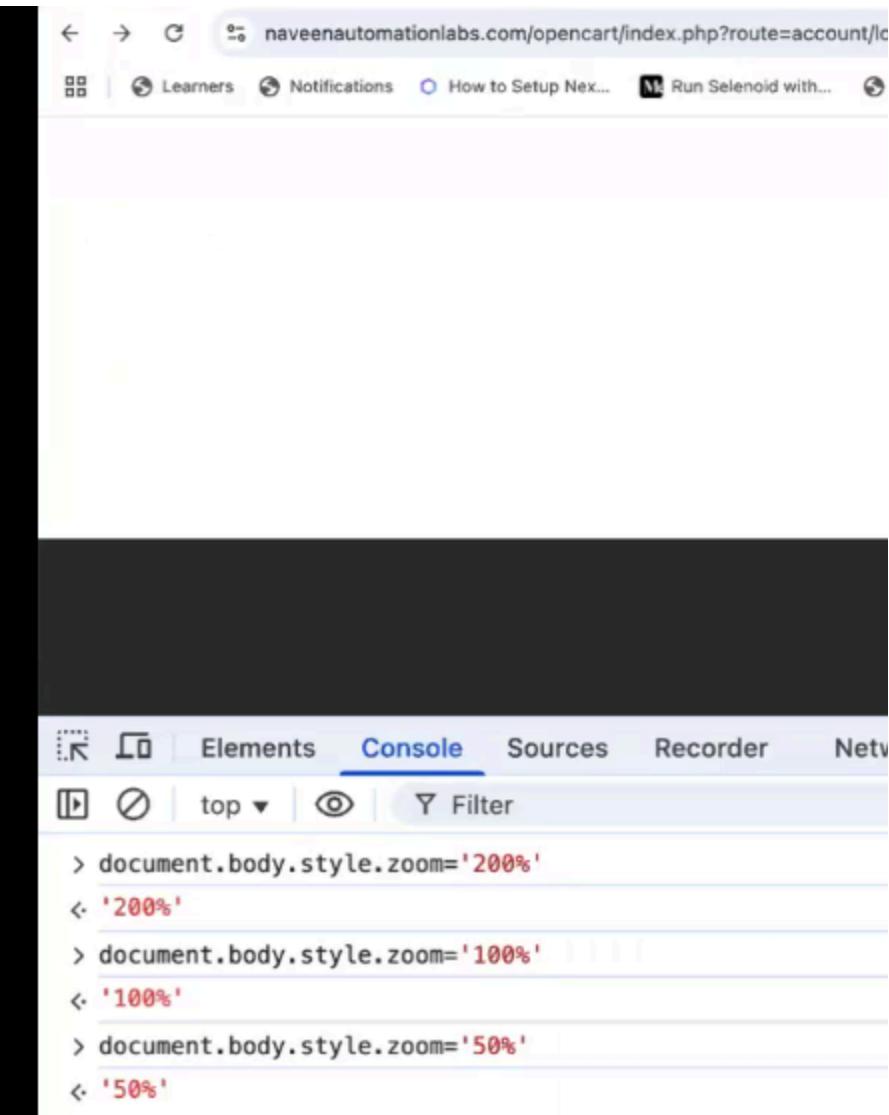


Zoom percent to 100-

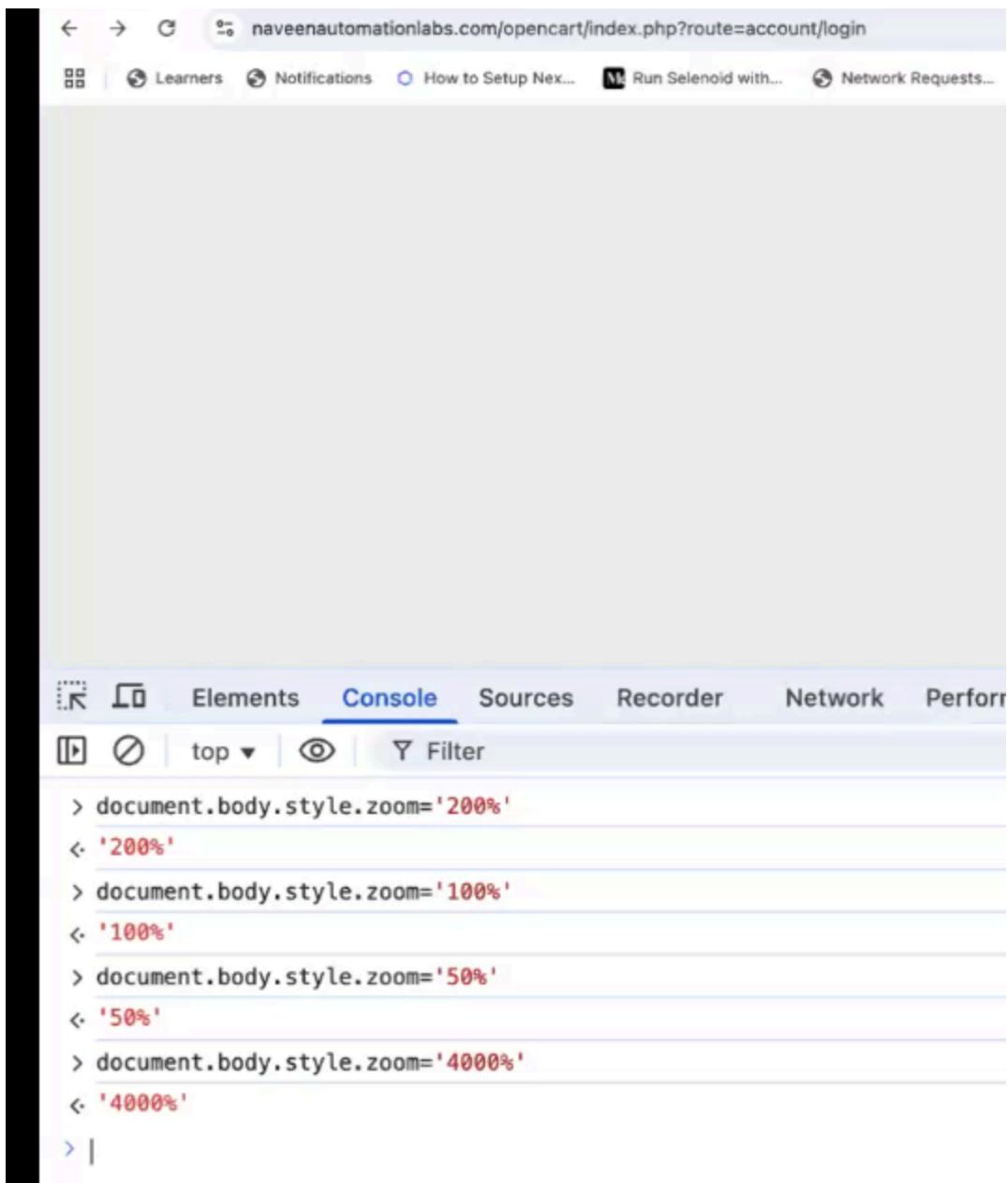
The screenshot shows a web browser window with the URL naveenautomationlabs.com/opencart/index.php?route=account/login. The browser's header includes links for 'Learners', 'Notifications', 'How to Setup Nex...', 'Run Selenoid with...', and 'Network Requests...'. Below the header, there is a message about tracking order status with a 'Continue' button. At the bottom of the browser window, the developer tools are open, specifically the 'Console' tab. The console shows the following JavaScript interactions:

```
> document.body.style.zoom='200%'  
< '200%'  
> document.body.style.zoom='100%'  
< '100%'  
>
```

50 percent zoom-



Zoom 4000 percent-

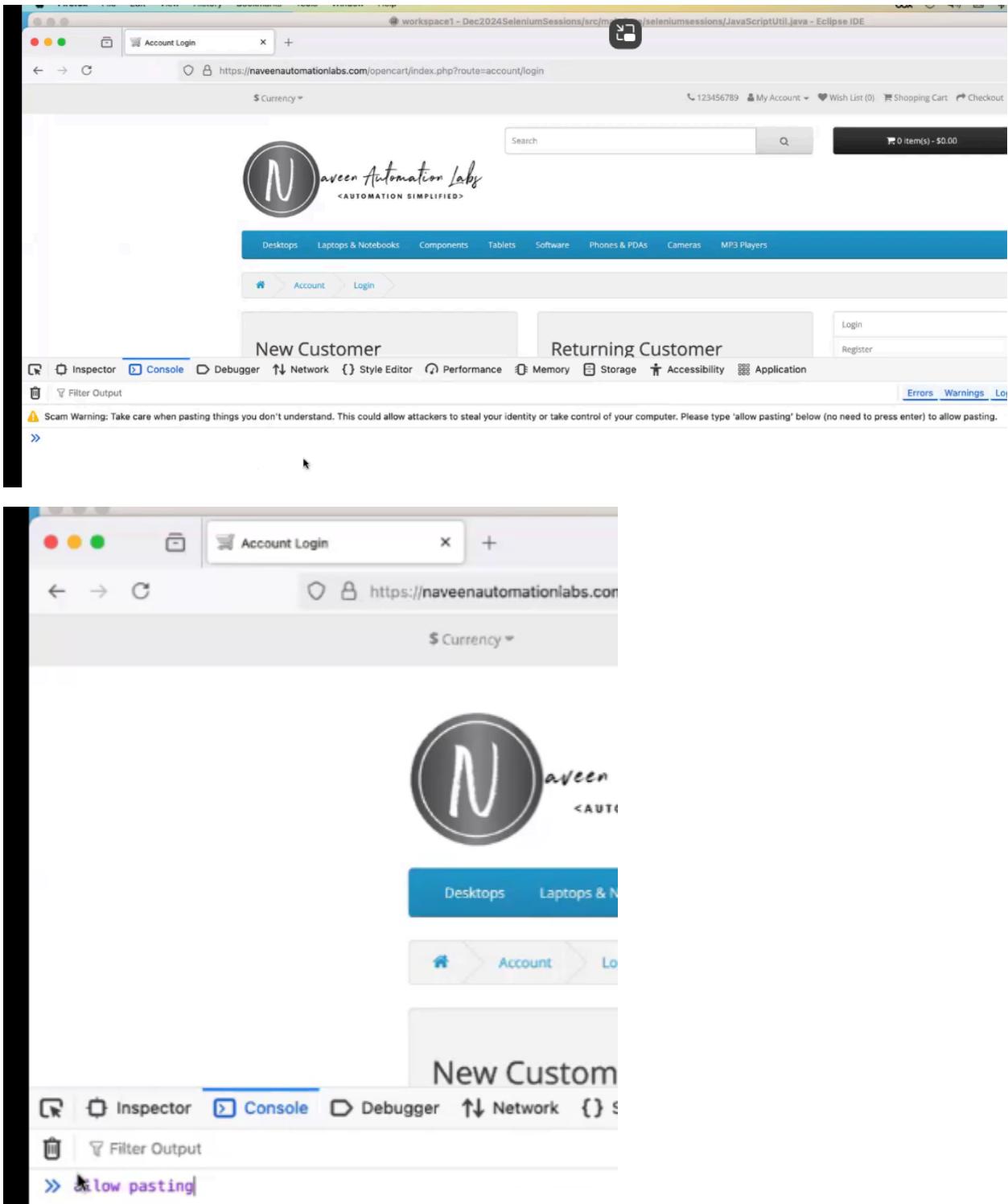


Zoom 400 percent-

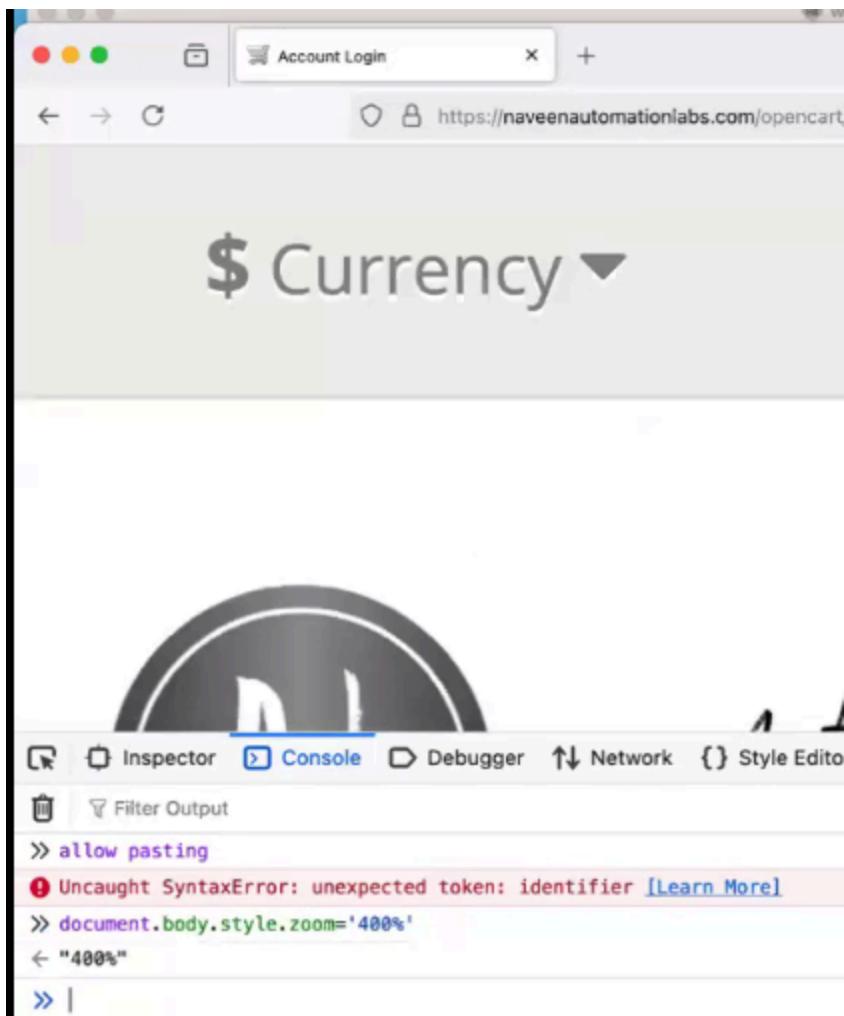


Check if working in firefox-

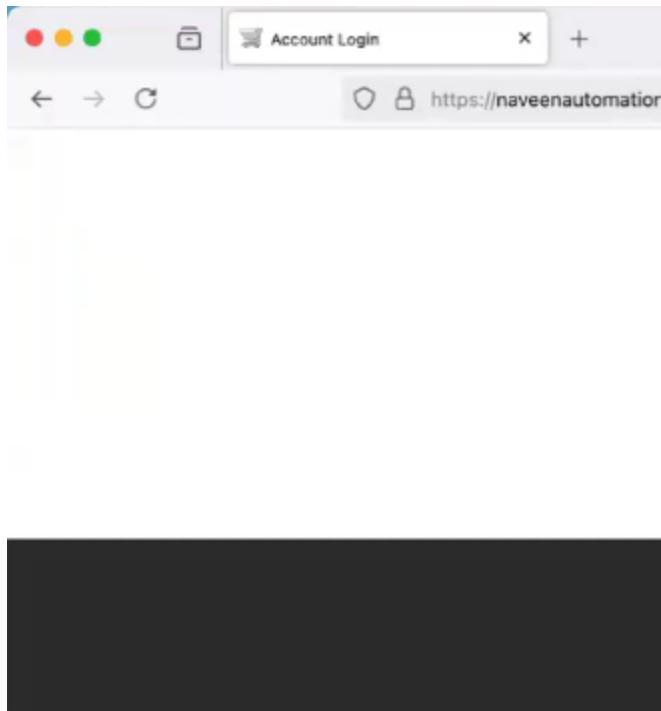
We get warning first time. Type the words as mentioned in the warning.



Zoom 400-



Zoom 50-

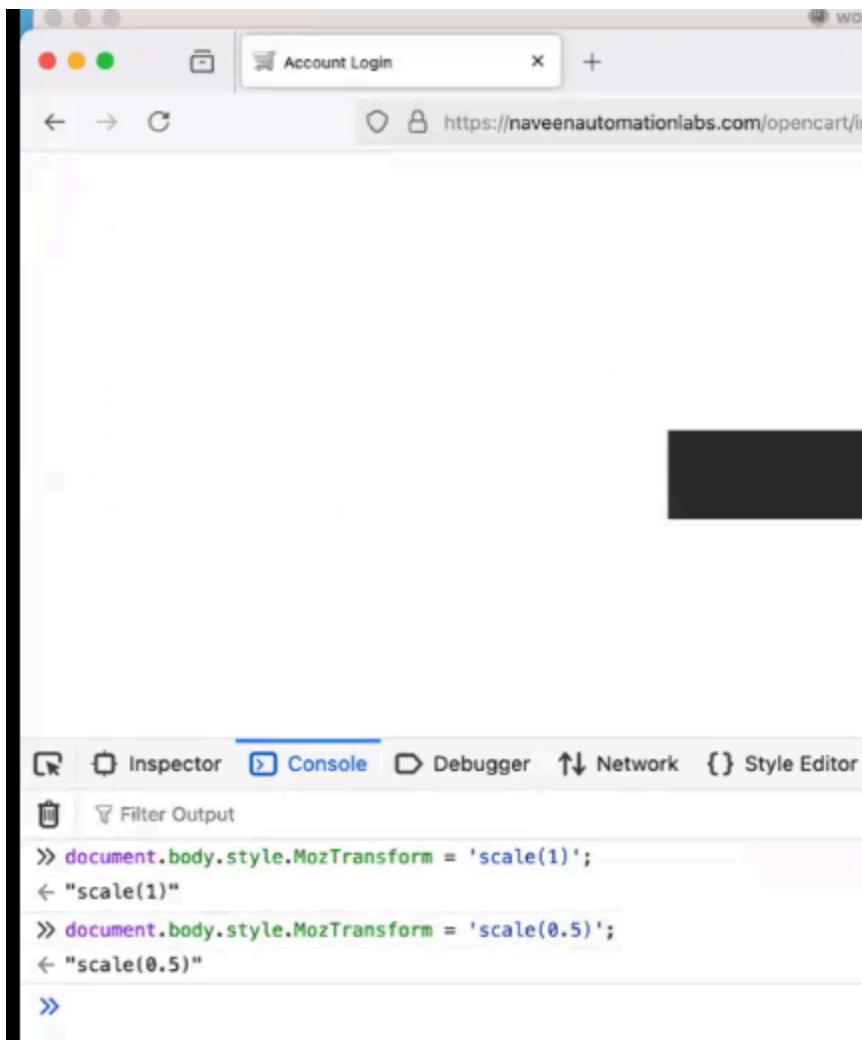
A screenshot of the Firefox developer tools Console tab. The tabs at the top are Inspector, Console (which is selected), Debugger, and Network. Below the tabs, there is a "Filter Output" button. The console output shows:

```
» allow pasting
✖ Uncaught SyntaxError: unexpected token: identifier [L
» document.body.style.zoom='400%
← "400%
» document.body.style.zoom='50%
← "50%
» |
```

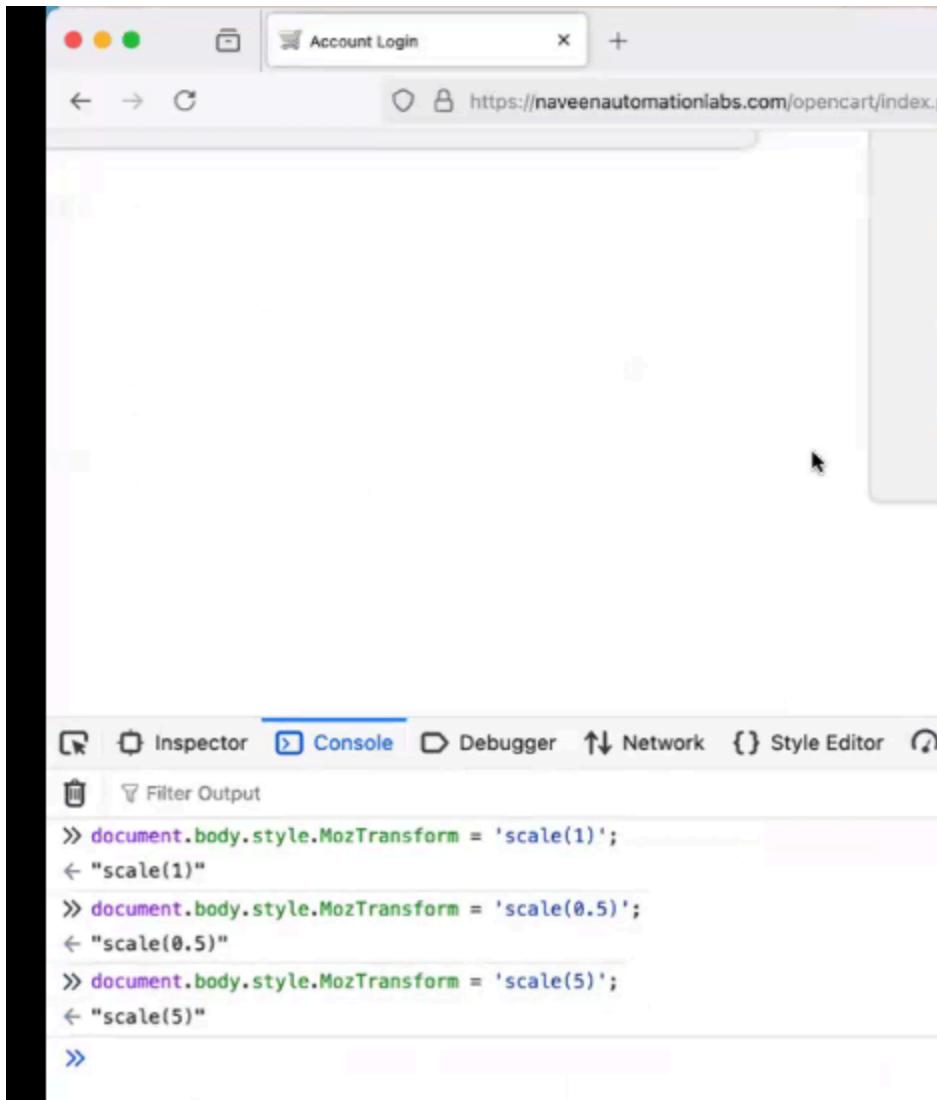
The error message "Uncaught SyntaxError: unexpected token: identifier [L" is highlighted in red.

Another way in ff-

In ff we have another method to zoom also.



Scale is 5-



Scale to 1-
1 means 100 percent.
0.5 means 50 percent.

The screenshot shows a web browser window with the URL <https://naveenautomationlabs.com/opencart/index.php?route=account/login>. The page title is "Account Login". The header includes the Naveen Automation Labs logo and navigation links for Catalog, Customer & Downloads, Components, Help, Support, Privacy & Policy, Contact, and API Docs.

The main content area has two sections:

- New Customer:** A form for creating a new account. It includes fields for First Name, Last Name, Email Address, Password, and Confirm Password, along with a "Continue" button.
- Returning Customer:** A form for logging in. It includes fields for Email Address and Password, along with a "Login" button.

Below the browser window, the developer tools console is open, showing the following JavaScript code being run:

```
document.body.style.MozTransform = 'scale(1)';
<"scale(1)">
document.body.style.MozTransform = 'scale(0.5)';
<"scale(0.5)">
document.body.style.MozTransform = 'scale(5)';
<"scale(5)">
document.body.style.MozTransform = 'scale(1)';
<"scale(1)">
```

Scale to 2-

The screenshot shows a web browser window with the following details:

- Title Bar:** Account Login
- URL:** https://naveenautomationlabs.com/opencart/index.php?route=account/login
- Main Content:** A "New Customer" registration form with a "Register Account" section. It explains the benefits of creating an account and has a "Continue" button.
- Right Sidebar:** A "Returning" user login sidebar with fields for E-Mail Address (naveenanimation), Password, and a "Login" button.
- Bottom:** Developer tools with tabs like Inspector, Console, Debugger, Network, Style Editor, Performance, Memory, and Storage. The Console tab is active, showing the following JavaScript code:

```
>> document.body.style.MozTransform = 'scale(1)';
← "scale(1)"
>> document.body.style.MozTransform = 'scale(0.5)';
← "scale(0.5)"
>> document.body.style.MozTransform = 'scale(5)';
← "scale(5)"
>> document.body.style.MozTransform = 'scale(1)';
← "scale(1)"
>> document.body.style.MozTransform = 'scale(2)';
← "scale(2"
>>
```

Js util class-
Add all zoom here.

```

91
92    /**
93     * example: "document.body.style.zoom = '400.0%'"
94     * @param zoomPercentage
95     */
96    public void zoomChromeEdgeFirefox(String zoomPercentage) {
97        String zoom = "document.body.style.zoom = '"+zoomPercentage+"%'";
98        js.executeScript(zoom);
99    }
100
101   /**
102    * example: "document.body.style.MozTransform = 'scale(0.5)'; "
103    * @param zoomPercentage
104    */
105    public void zoomFirefox(String zoomPercentage) {
106        String zoom = "document.body.style.MozTransform = 'scale("+zoomPercentage+")'";
107        js.executeScript(zoom);
108    }
109
110

```

Click method using js- Js util class.

```

111
112    public void clickElementByJS(WebElement element) {
113        js.executeScript("arguments[0].click();", element);
114    }
115

```

Naveen website.

Register page.

Click agree checkbox.

Test class-

Class name-JavaScriptExecutorConcept

Changed url.

```

13
14     WebDriver driver = new ChromeDriver();
15     driver.get("https://naveenautomationlabs.com/opencart/index.php?route=account/register");
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63     WebElement agree = driver.findElement(By.name("agree"));
64     jsUtil.clickElementByJS(agree);
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110

```

Register Account

naveenautomationlabs.com/opencart/index.php?route=account/register

Chrome is being controlled by automated test software.

If you already have an account with us, please login at the [login page](#).

Your Personal Details

* First Name	First Name
* Last Name	Last Name
* E-Mail	E-Mail
* Telephone	Telephone

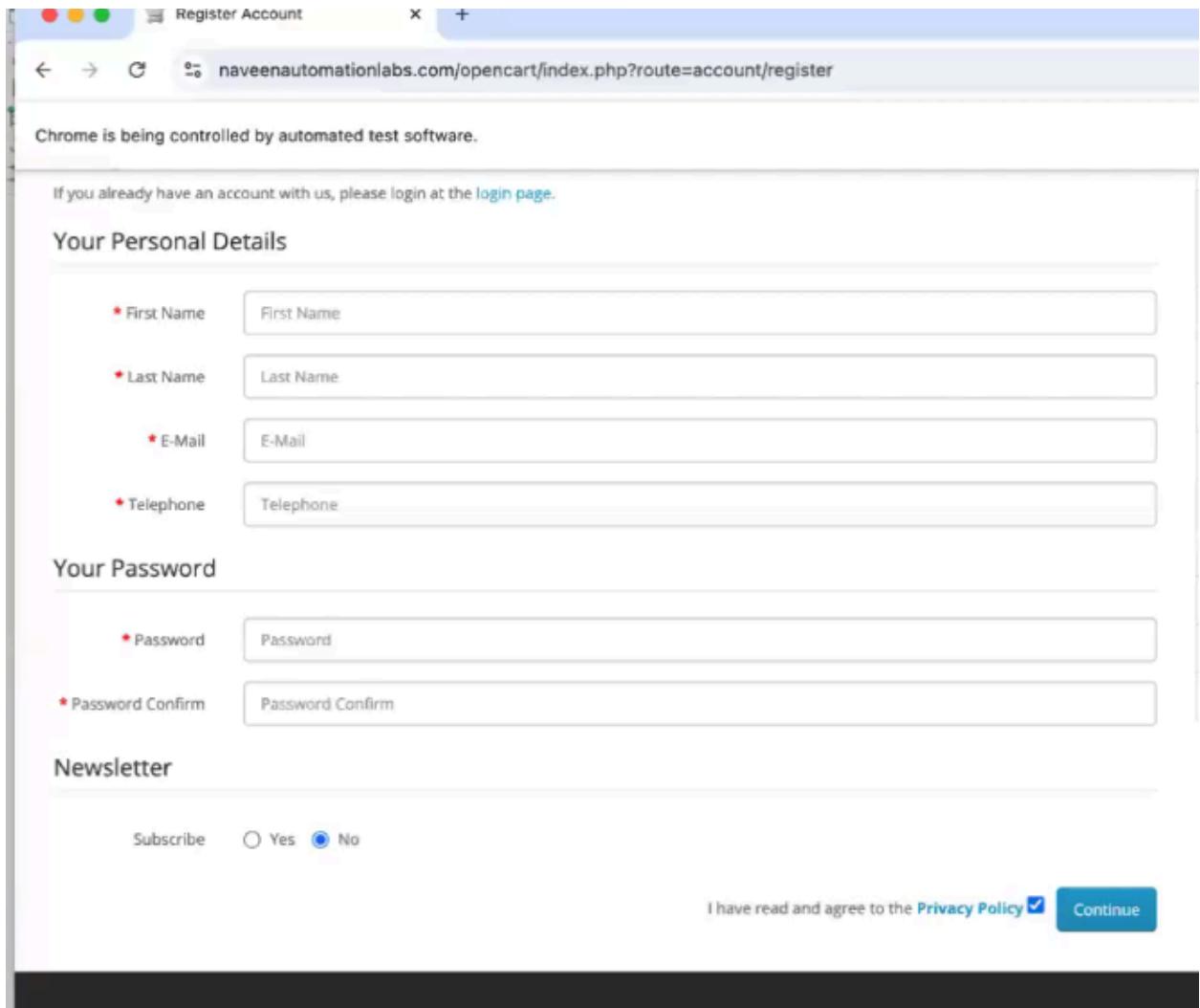
Your Password

* Password	Password
* Password Confirm	Password Confirm

Newsletter

Subscribe Yes No

I have read and agree to the [Privacy Policy](#) [Continue](#)



Use js when nothing else works.
It uses dom and backdoor entry to perform operations on browser.
Send keys using js-

The screenshot shows a web browser window with the URL naveenautomationlabs.com/opencart/index.php?route=account/register. The page title is "Register Account". On the left, there is a sidebar with links like Login, Register, Forgotten Password, My Account, Address Book, Wish List, Order History, Downloads, Recurring payments, Reward Points, Returns, Transactions, and Newsletter. The main form has fields for First Name, Last Name, E-Mail, Telephone, and Password, each with validation stars. The "First Name" field is currently selected and highlighted with a green border. In the bottom-left corner of the browser, the developer tools are open, with the "Console" tab selected. The console output shows an uncaught promise error:

```
Uncaught (in promise) Error: A listener indicated an asynchronous response by returning true, but the message channel closed
> document.getElementById('input-firstname');
<  <input type="text" name="firstname" value placeholder="First Name" id="input-firstname" class="form-control">
>
```

Get element by id method - returns web element and gives html of that element.
How to enter value using js-

The screenshot shows a web browser window with the URL naveenautomationlabs.com/opencart/index.php?route=account/register. The page title is "Register Account". On the right side, there is a sidebar with links such as Login, Register, Forgotten Password, My Account, Address Book, Wish List, Order History, Downloads, Recurring payments, Reward Points, Returns, Transactions, and Newsletter. Below the sidebar is a "Your Personal Details" form with fields for First Name (containing "testing"), Last Name, E-Mail, Telephone, Password, and Password Confirm. The developer console at the bottom is open to the "Console" tab, showing the following JavaScript error and stack trace:

```
✖ Uncaught (in promise) Error: A listener indicated an asynchronous response by returning true, but the message channel closed
> document.getElementById('input-firstname');
<   <input type="text" name="firstname" value placeholder="First Name" id="input-firstname" class="form-control">
> document.getElementById('input-firstname').value='testing'
< 'testing'
> |
```

To use css style locator-
Use query selector in js.

The screenshot shows a web browser window with three tabs: "Register Account", "Free CRM software for customer relationship management", and "Online Shopping site in India". The "Register Account" tab is active, displaying a registration form titled "Register Account". The form has sections for "Your Personal Details" and "Your Password". The "Console" tab of the developer tools is selected, showing the following JavaScript error:

```
Uncaught (in promise) Error: A listener indicated an asynchronous response by returning true, but the message channel closed before the response was received.
```

The developer tools also show the DOM structure for the first name input field:

```
> document.getElementById('input-firstname');
<  <input type="text" name="firstname" value placeholder="First Name" id="input-firstname" class="form-control">
> document.getElementById('input-firstname').value='testing'
< 'testing'
> document.querySelector('#input-firstname');
<  <input type="text" name="firstname" value placeholder="First Name" id="input-firstname" class="form-control">
>
```

Enter value in firstname-

The screenshot shows a web browser window with the URL naveenautomationlabs.com/opencart/index.php?route=account/register. The page title is "Register Account". On the right side, there is a sidebar with links such as Login, Register, Forgotten P..., My Account, Address Book, Wish List, Order History, Downloads, Recurring profiles, Reward Points, Returns, Transaction history, and Newsletter. Below the sidebar, the developer tools console tab is selected, showing the following error message:

```
x Uncaught (in promise) Error: A listener indicated an asynchronous response by returning true, but the message channel
> document.getElementById('input-firstname');
<   <input type="text" name="firstname" value placeholder="First Name" id="input-firstname" class="form-control">
> document.getElementById('input-firstname').value='testing'
< 'testing'
> document.querySelector('#input-firstname');
<   <input type="text" name="firstname" value placeholder="First Name" id="input-firstname" class="form-control">
> document.querySelector('#input-firstname').value='naveen'
< 'naveen'
> |
```

If you want by name-

It returns multiple elements.

So return type is array list. In js it becomes node list.

Get length.

The screenshot shows a web browser window with three tabs: 'Register Account', 'Free CRM software for customer relationship management', and 'Online Shopping site in India'. The 'Register Account' tab is active, displaying a registration form. The form has sections for 'Your Personal Details' and 'Your Password'. Fields include First Name (naveen), Last Name, E-Mail, Telephone, Password, and Password Confirm. Below the form, the browser's developer tools console is open, showing the following JavaScript code:

```

> document.getElementsByName('firstname')
< NodeList [input#input-firstname.form-control] i
  > 0: input#input-firstname.form-control
    length: 1
  > [[Prototype]]: NodeList
> document.getElementsByName('firstname').length
< 1
>

```

Jsutil class-

Add the send keys method.

```

116
117=   public void sendKeysUsingWithIdByJS(String id, String value) {
118     js.executeScript("document.getElementById('" + id + "').value='" + value + "'");
119     //document.getElementById('input-email').value ='tom@gmail.com'
120   }
121

```

Xpath replacement in js-

Evaluate is the function.

```

> document.evaluate("//input[@id='input-email']", document, null, XPathResult.ANY_TYPE, null);
<-> XPathResult {resultType: 4, invalidIteratorState: false}
    booleanValue: ...
    invalidIteratorState: false
    numberValue: ...
    resultType: 4
    singleNodeValue: ...
    snapshotLength: ...
    stringValue: ...
  [[Prototype]: XPathResult]

```

Removed the js executor initialisation line as it was repeated and already initialised in constructor-Js util class.

From some of the methods.

```

50
51  public void scrollPageDown() {
52      js.executeScript("window.scrollTo(0, document.body.scrollHeight)");
53  }
54
55  public void scrollPageDown(String height) {
56      js.executeScript("window.scrollTo(0, '" + height + "')");
57  }
58
59  public void scrollPageUp() {
60      js.executeScript("window.scrollTo(document.body.scrollHeight, 0)");
61  }
62
63  public void scrollIntoView(WebElement element) {
64      js.executeScript("arguments[0].scrollIntoView(true);", element);
65  }
66
67  public void drawBorder(WebElement element) {
68      js.executeScript("arguments[0].style.border='3px solid red'", element);
69  }

```

```
79~ private void changeColor(String color, WebElement element) {  
80 |     js.executeScript("arguments[0].style.backgroundColor = '" + color + "'", element); //GBGBGBGBGBGB  
81  
82     try {  
83         Thread.sleep(20);  
84     } catch (InterruptedException e) {  
85     }  
86 }
```