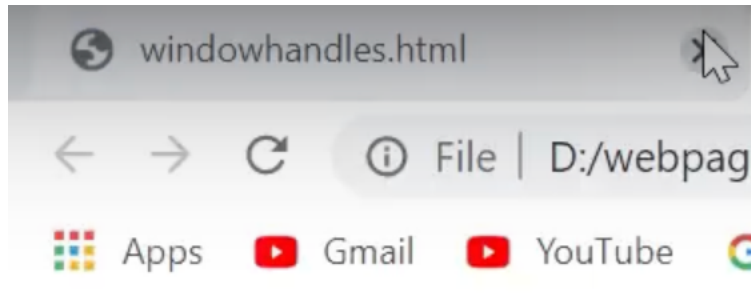Buradaki facebook , google ve wiki linklerine tikladigimizda yeni windows lar acilacak
Sms get title methoduyla current window u bulmak istiyoruz ama gostermiyor , driver hala
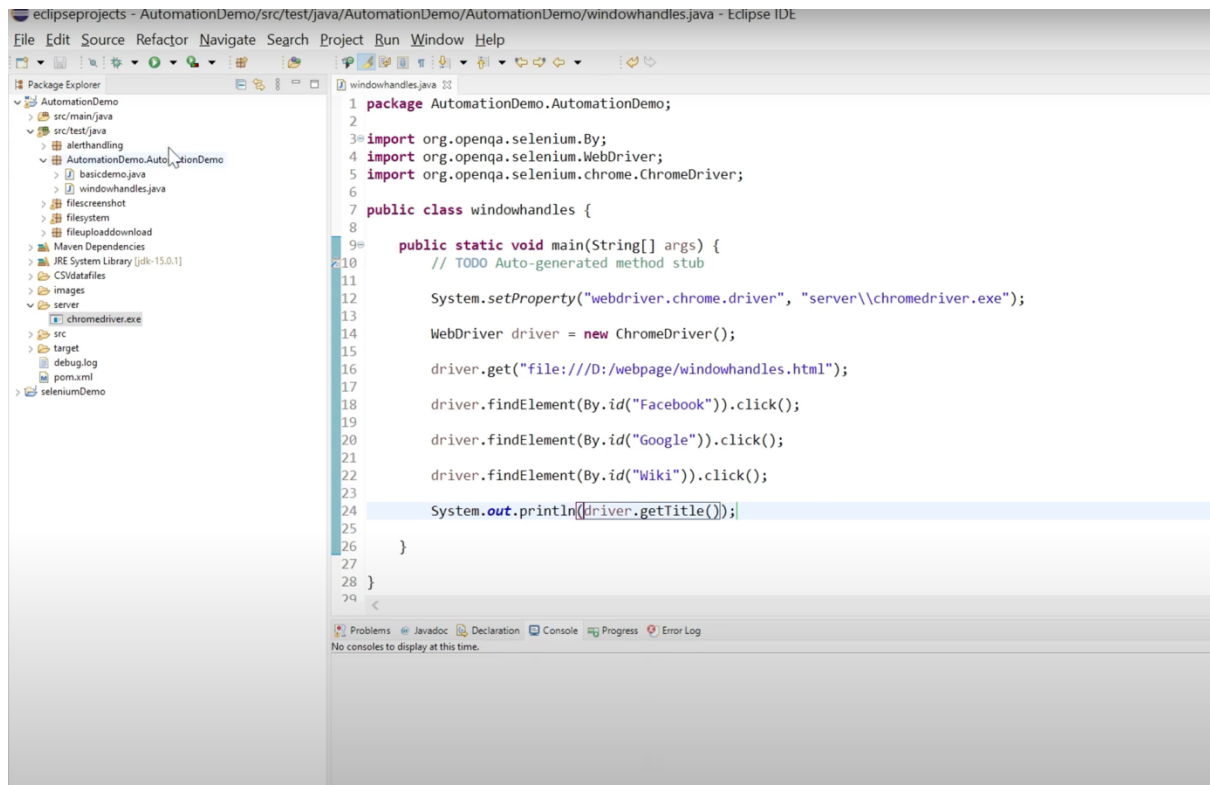ilk windowda


```java
package AutomationDemo.AutomationDemo;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class windowhandles {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        System.setProperty("webdriver.chrome.driver", "server\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();

        driver.get("file:///D:/webpage/windowhandles.html");

        driver.findElement(By.id("Facebook")).click();

        driver.findElement(By.id("Google")).click();

        driver.findElement(By.id("Wiki")).click();

        System.out.println(driver.getTitle());

    }

}
```

Herhangi bir title vermedi cunku hala ilk windowda selenium focus on ,diger windowlara
switch etmeliyiz
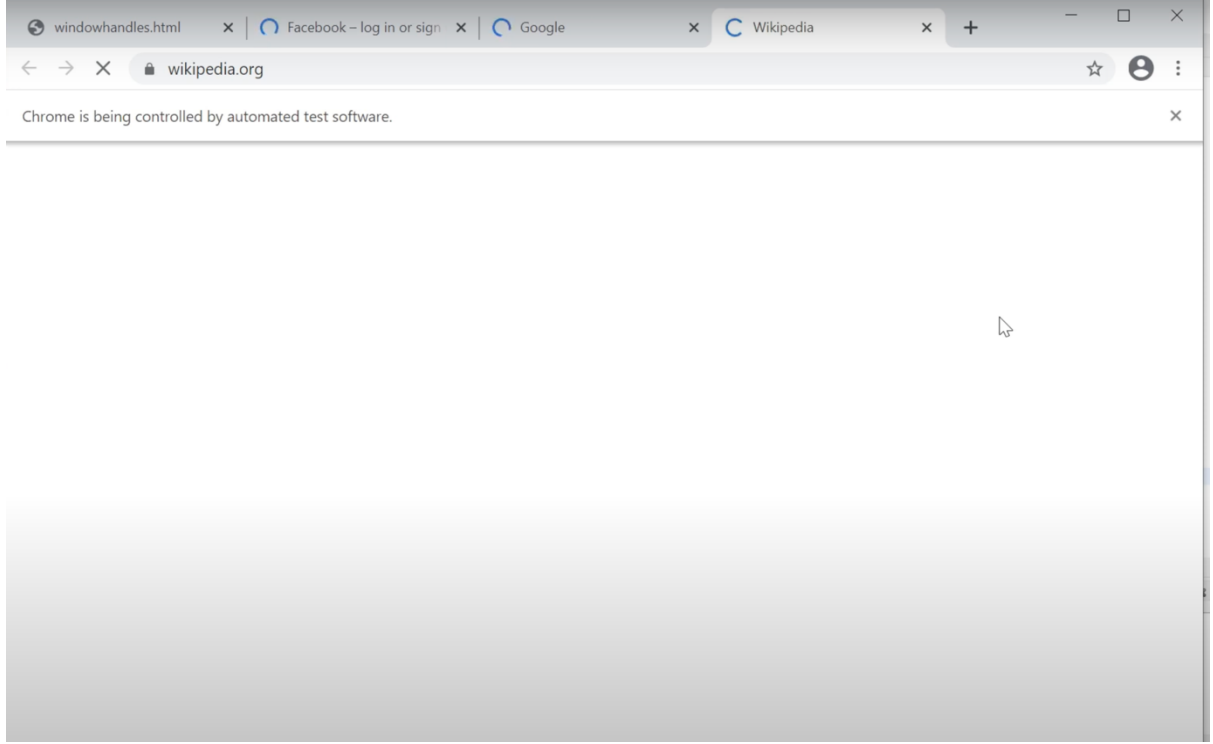
Donguyu 1 indexi ile baslatti, cunku close methodunun ilk baslangic window unu
kapatmasini istemedi

4 tane windows var , 3 u yeni acildi

```
System.out.println(driver.getTitle());

ArrayList<String> windows = new ArrayList(driver.getWindowHandles());

for(int i = 1; i< windows.size() ; i++) {

    driver.switchTo().window(windows.get(i));

    System.out.println("my current active window title : " +driver.getTitle());

    driver.close();

}

}

}
```
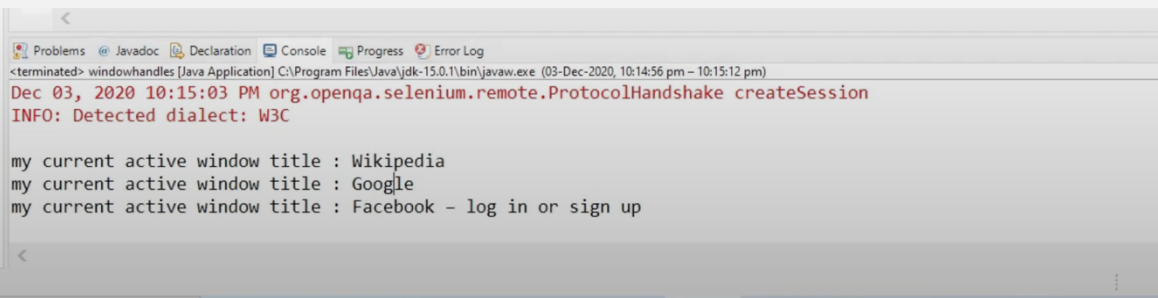
```
Problems  @ Javadoc  Declaration  Console  Progress  Error Log
<terminated> windowhandles [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe  (03-Dec-2020, 10:14:56 pm – 10:15:12 pm)
Dec 03, 2020 10:15:03 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C

my current active window title : Wikipedia
my current active window title : Google
my current active window title : Facebook – log in or sign up
```

Interview da nasil multiple windows lari handle ediyorsun derlerse

Storing them in an arraylist and then we are iterating one by one

Selenium Alert & Popup Window Handling: How to Handle?

By[Krishna Rungta](#)UpdatedAugust 20, 2022

In this tutorial, we will learn how to handle popup in Selenium and different types of alerts found in web application [Testing](#). We will also see how to handle Alert in Selenium WebDriver and learn how do we accept and reject the alert depending upon the alert types.
In this tutorial, you will learn-

- [What is Alert?](#)
- [How to handle Alert in Selenium WebDriver](#)
- [How to handle Selenium Popup window using Webdriver](#)

What is Alert in Selenium?

An **Alert in Selenium** is a small message box which appears on screen to give the user some information or notification. It notifies the user with some specific information or error, asks for permission to perform certain tasks and it also provides warning messages as well.
Here are few alert in Selenium types:

1) Simple Alert

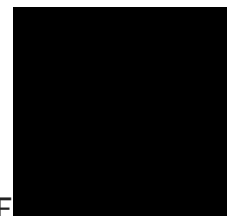The simple alert class in Selenium displays some information or warning on the screen.
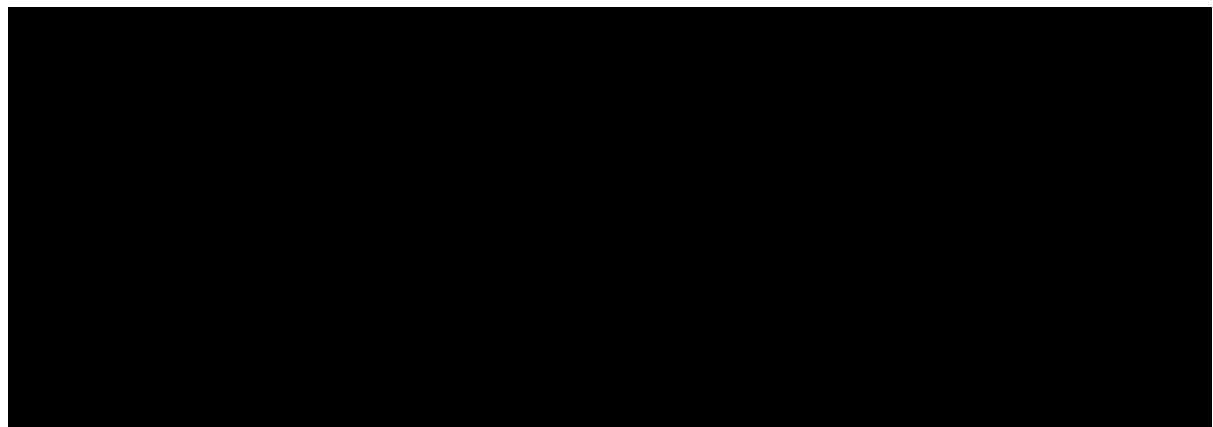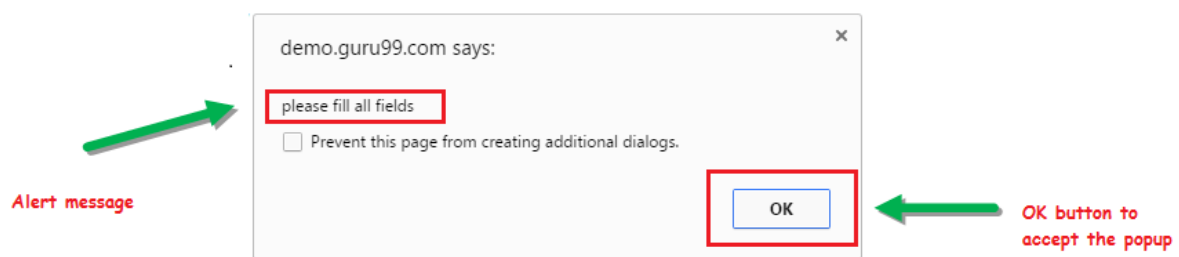
Error! Filename not specified.

02:5435:04   EXPLORE MORE

Learn Java Programming with Beginners Tutorial08:32   Linux
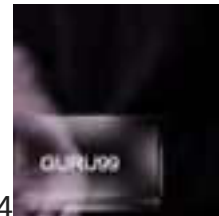
Tutorial for Beginners: Introduction to Linux Operating...01:35
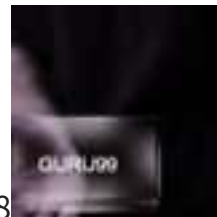
What is Integration Testing Software Testing Tutorial03:04   What is

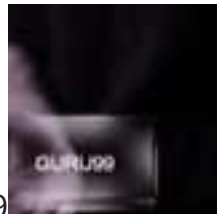JVM (Java Virtual Machine) with Architecture JAVA...02:24   How to

write a TEST CASE Software Testing Tutorial01:08   Seven Testing
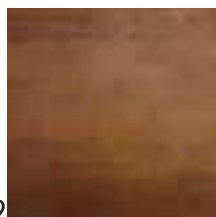
Principles Software Testing05:01   Linux File Permissions

Commands with Examples13:29   How to use Text tool in
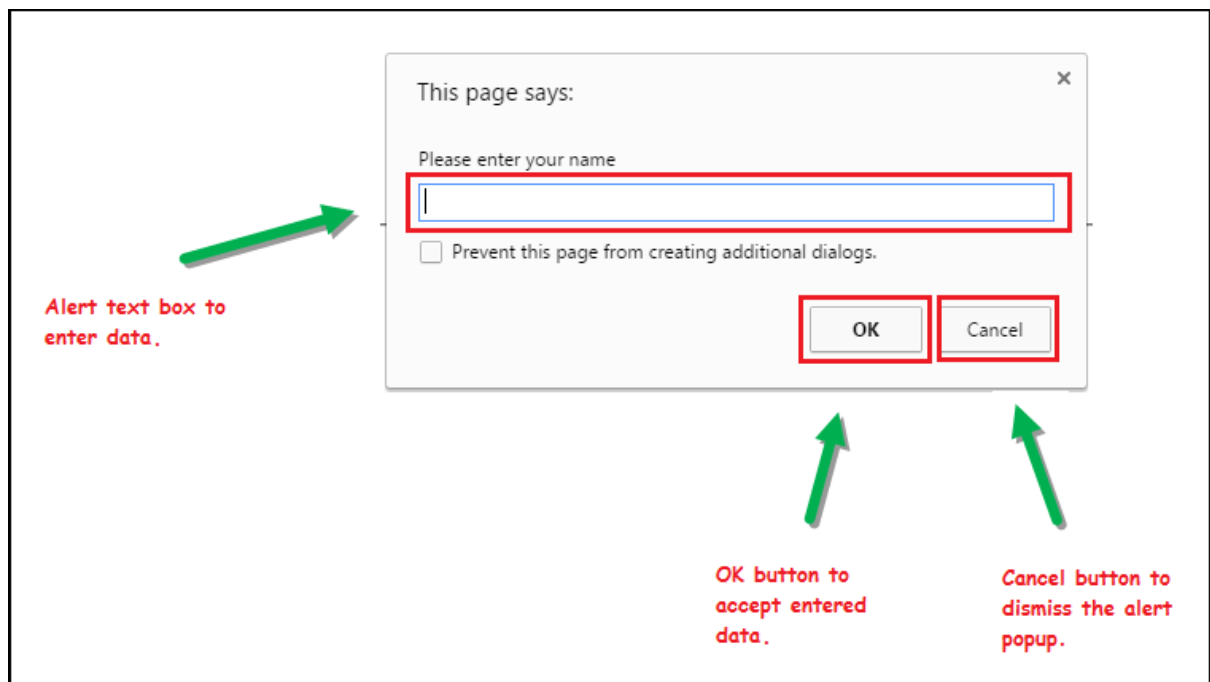
Photoshop CC Tutorial08:32   What is NoSQL Database
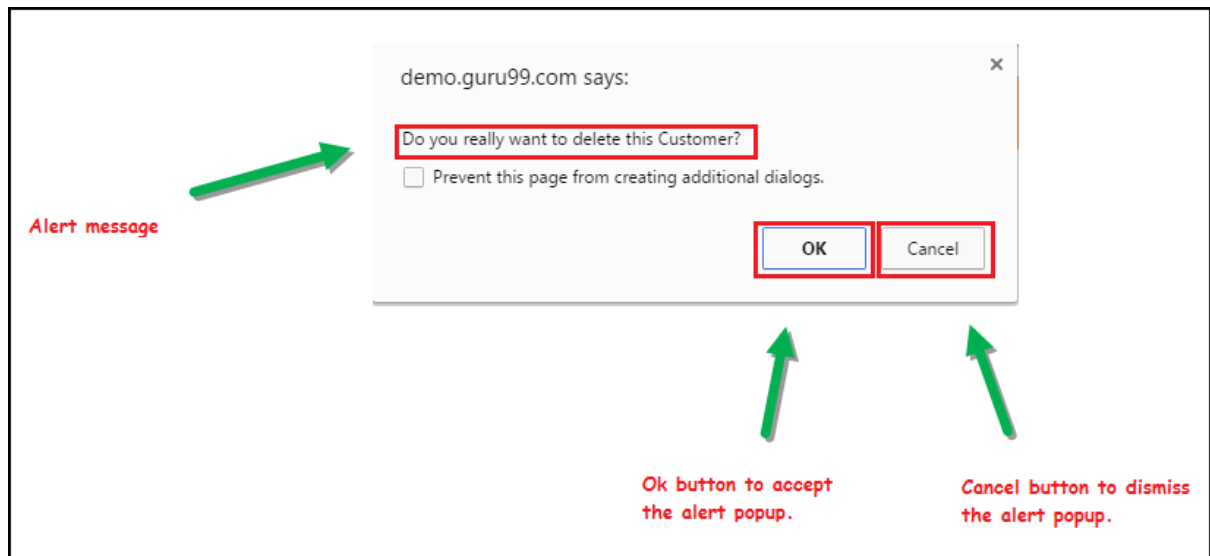
2) Prompt Alert.

This Prompt Alert asks some input from the user and Selenium webdriver can enter the text using sendkeys(" input…. ").

3) Confirmation Alert.

This confirmation alert asks permission to do some type of operation.

How to handle Alert in Selenium WebDriver

Alert interface provides the below few methods which are widely used in [Selenium Webdriver](#).

1) void dismiss() **// To click on the 'Cancel' button of the alert.**

```
driver.switchTo().alert().dismiss();
```
2) void accept() **// To click on the 'OK' button of the alert.**

```
driver.switchTo().alert().accept();
```
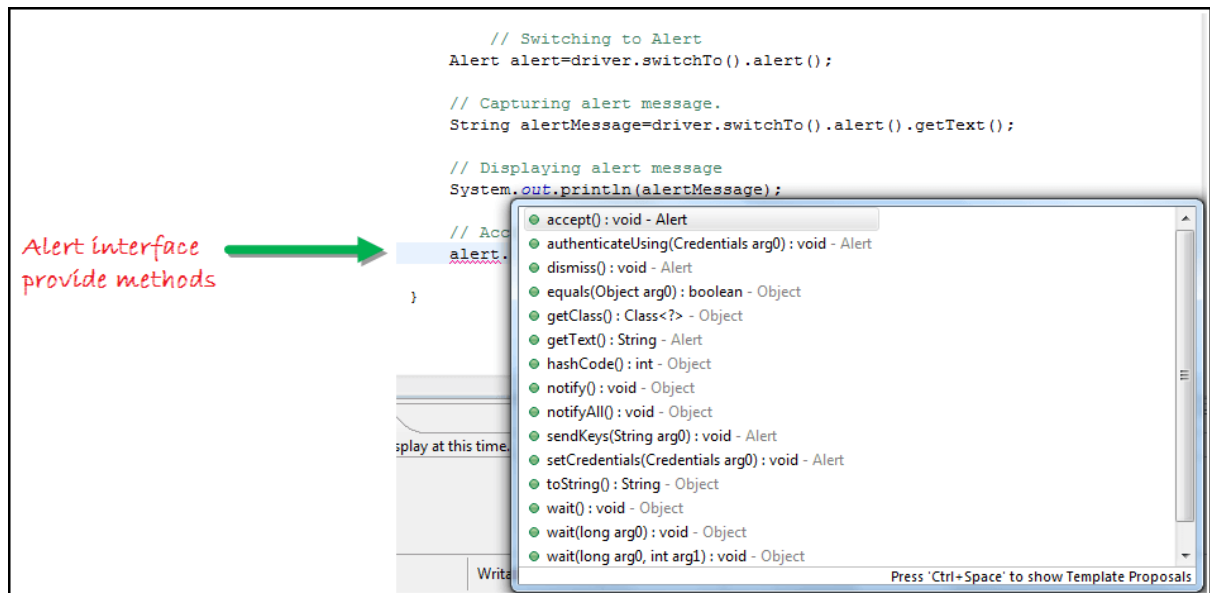3) String getText**() // To capture the alert message.**

```
driver.switchTo().alert().getText();
```
4) void sendKeys(String stringToSend) **// To send some data to alert box.**

```
driver.switchTo().alert().sendKeys("Text");
```

You can see a number of Alert methods are displayed as shown in below screen suggested by Eclipse.
We can easily switch to alert from the main window by using Selenium's **.switchTo()** method.

```
            // Switching to Alert
    Alert alert=driver.switchTo().alert();

            // Capturing alert message.
    String alertMessage=driver.switchTo().alert().getText();

            // Displaying alert message
    System.out.println(alertMessage);
```

Alert interface provide methods →

Now we automate the given below scenario.

In this scenario, we will use Guru99 demo site to illustrate Selenium Alert handling.

Step 1) Launch the web browser and open the site http://demo.guru99.com/test/delete_customer.php
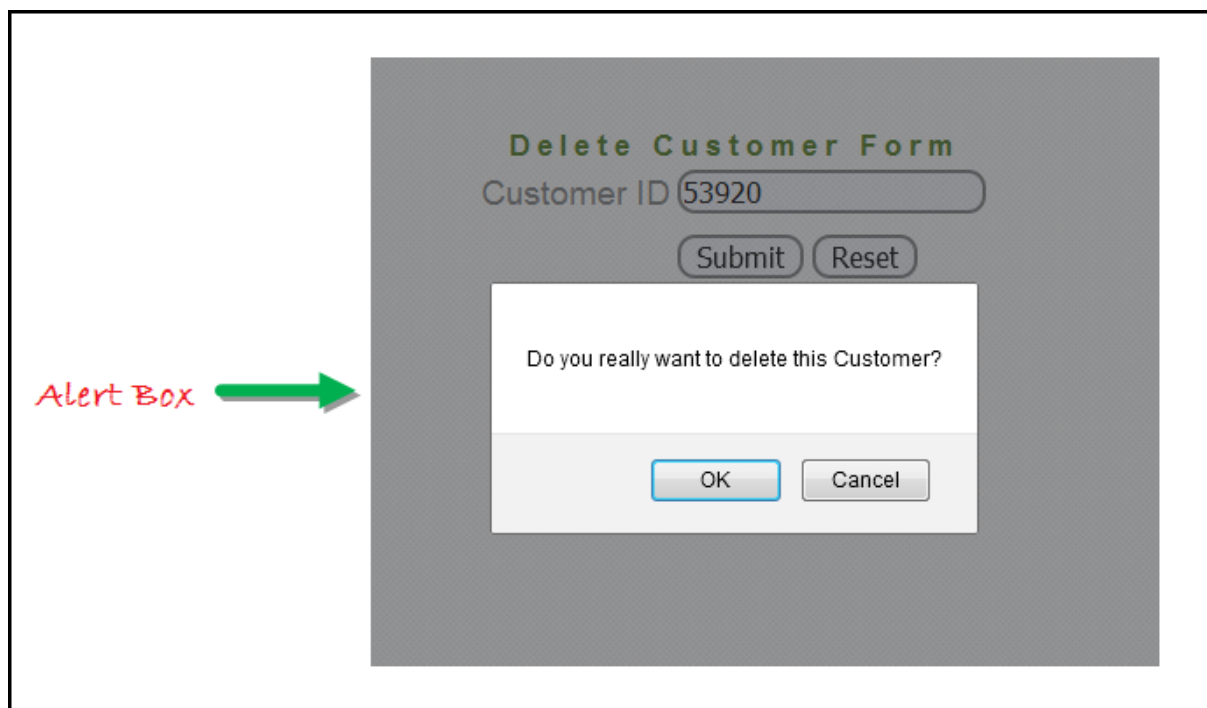
Step 2) Enter Any Customer id.

Step 3) After entering the customer ID, Click on the "Submit" button.

Step 4) Reject/accept the alert.

Handling Alert in Selenium Webdriver using above scenario

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.NoAlertPresentException;
import org.openqa.selenium.Alert;

public class AlertDemo {
```

```
        public static void main(String[] args) throws
NoAlertPresentException,InterruptedException  {


        System.setProperty("webdriver.chrome.driver","G:\\chromedriver.exe"
);
            WebDriver driver = new ChromeDriver();


        // Alert Message handling

        driver.get("http://demo.guru99.com/test/delete_customer.php");


        driver.findElement(By.name("cusid")).sendKeys("53920");

        driver.findElement(By.name("submit")).submit();


        // Switching to Alert
        Alert alert = driver.switchTo().alert();

        // Capturing alert message.
        String alertMessage= driver.switchTo().alert().getText();


        // Displaying alert message
        System.out.println(alertMessage);
        Thread.sleep(5000);

        // Accepting alert
        alert.accept();
    }

}
```

Output :

When you execute the above code, it launches the site. Try to delete Customer ID by handling confirmation alert that displays on the screen, and thereby deleting customer id from the application.

How to handle Selenium Pop-up window using Webdriver

In automation, when we have multiple windows in any web application, the activity may need to switch control among several windows from one to other in order to complete the operation. After completion of the operation, it has to return to the main window i.e. parent window in Selenium. We will see this further in the article with an example.

In Selenium web driver there are methods through which we can handle multiple windows.

Driver.getWindowHandles();

To handle all opened windows by web driver, we can use "Driver.getWindowHandles()" and then we can switch window from one window to another in a web application. Its return type is Iterator<String>.

Driver.getWindowHandle();

When the site opens, we need to handle the main window by **driver.getWindowHandle()**. This will handle the current window that uniquely identifies it within this driver instance. Its return type is String.
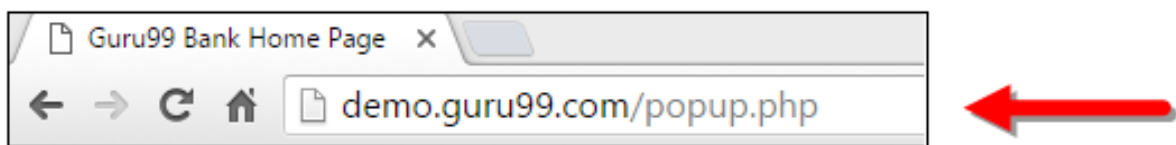
For Window handling in Selenium, we will follow the below steps:

Now, we will automate the given below scenario to see how to handle multiple windows using Selenium Webdriver.
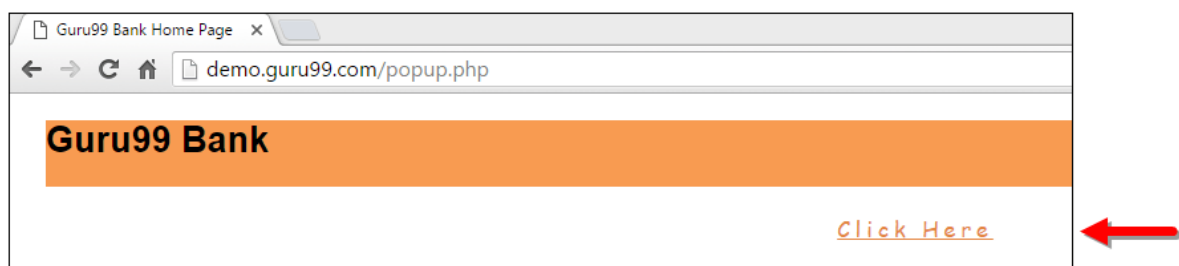In this scenario, we will use "Guru99" demo site to illustrate window handling.

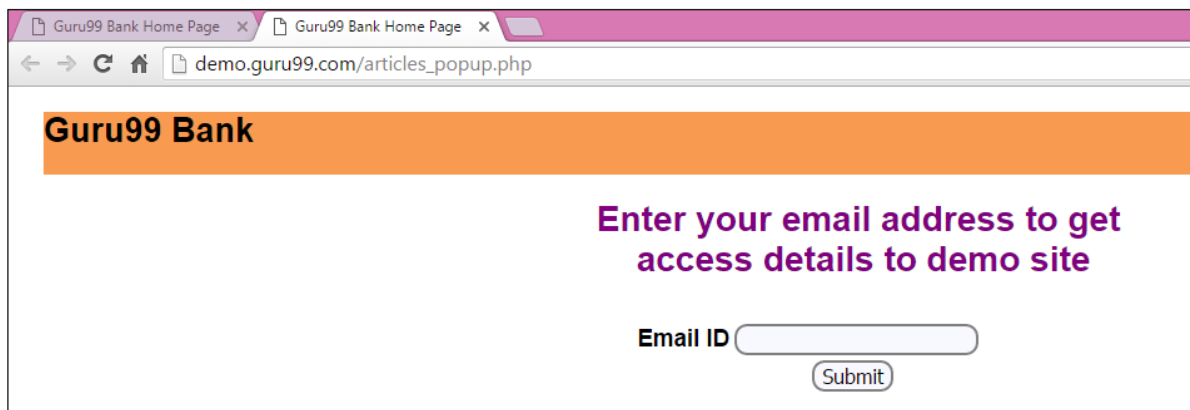**Step 1)** Launch the site.

Launch the browser and open the site `http://demo.guru99.com/popup.php`

Step 2) Click on link "Click Here ".

When the user clicks on the " Click Here " link, new child window opens.
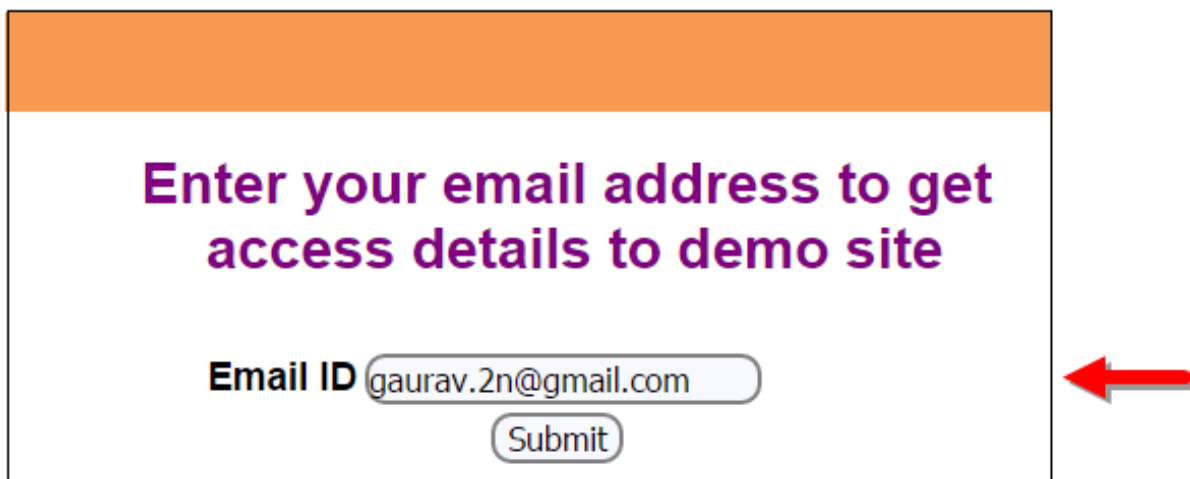
Step 3) New Child Window opens.

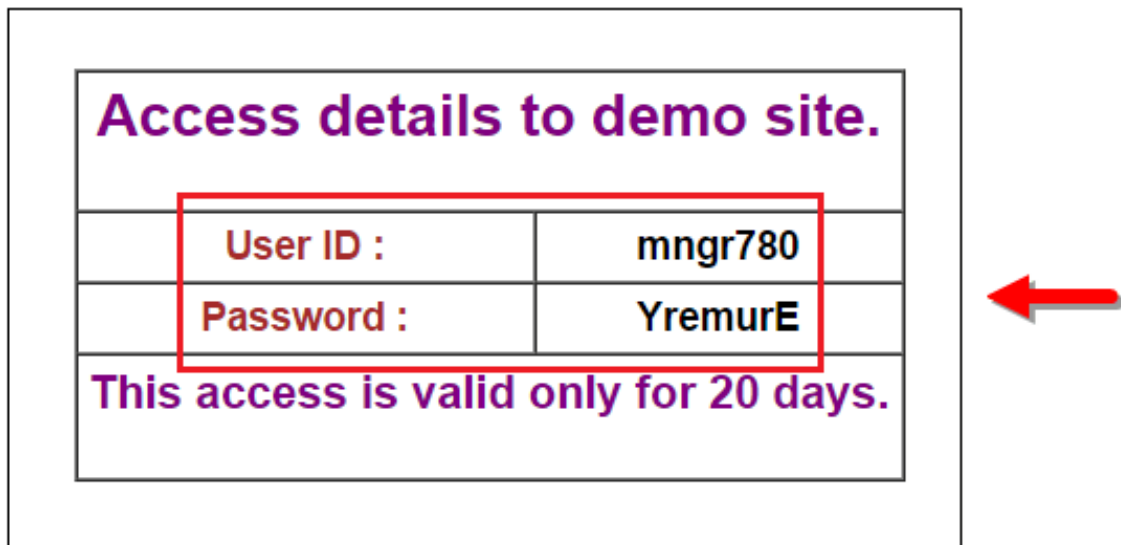A new window opens, ask the user to enter email id and submit the page.

Step 4) Enter your email ID and submit.

Step 5) Display the Access Credentials on submitting the page.

Access details to demo site.

User ID :	mngr780
Password :	YremurE

This access is valid only for 20 days.
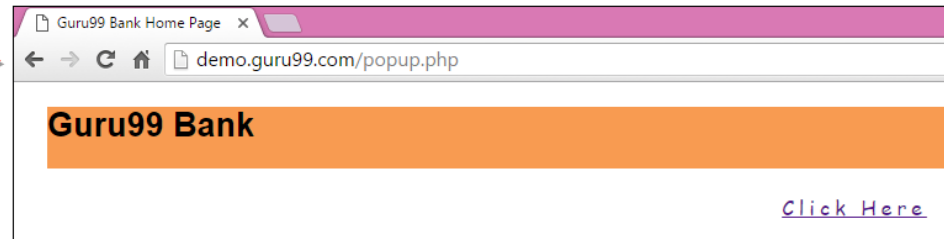
When you execute the code, you will see the child window is open in new tab.

1. Close the Child window on which credentials are displayed.

Close this Child Window

Guru99 Bank Home Page ✕   Guru99 Bank Home Page ✕

demo.guru99.com/articles_popup.php

Guru99 Bank

2. Switch to the parent window.

Parent Window

Handling multiple windows in Selenium webdriver using above scenario.

```java
import java.util.Iterator;
import java.util.Set;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class WindowHandle_Demo {

    public static void main(String[] args) throws InterruptedException {

            WebDriver driver=new FirefoxDriver();

        //Launching the site.
            driver.get("http://demo.guru99.com/popup.php");

        driver.manage().window().maximize();

driver.findElement(By.xpath("//*[contains(@href,'popup.php')]")).click();


        String MainWindow=driver.getWindowHandle();

        // To handle all new opened window.
            Set<String> s1=driver.getWindowHandles();
        Iterator<String> i1=s1.iterator();

        while(i1.hasNext())
        {
            String ChildWindow=i1.next();

            if(!MainWindow.equalsIgnoreCase(ChildWindow))

            {

                    // Switching to Child window
                    driver.switchTo().window(ChildWindow);
                    driver.findElement(By.name("emailid"))
                    .sendKeys("gaurav.3n@gmail.com");


                    driver.findElement(By.name("btnLogin")).click();


                    // Closing the Child Window.
                     driver.close();
            }
        }
        // Switching to Parent window i.e Main Window.
```
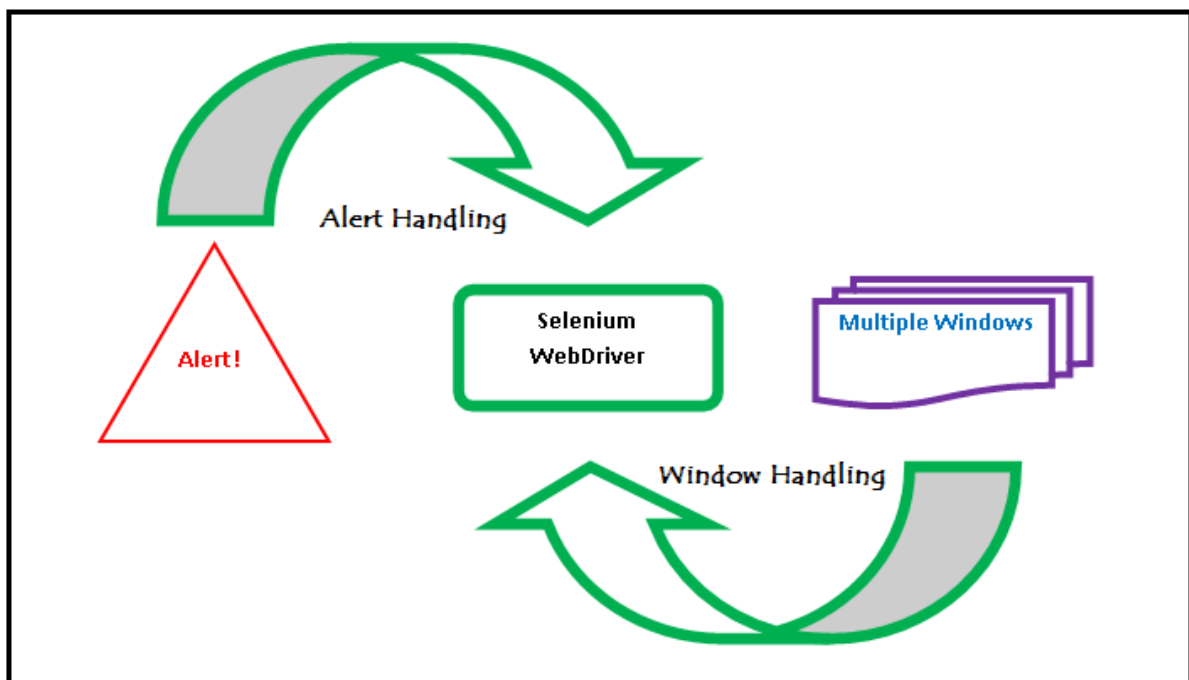
```
            driver.switchTo().window(MainWindow);

    }
}
```
Output:

When you execute the above code, it launches the site and on clicking the link "Click here," it opens up a child window in a new tab. You can close the child window, and switch to the parent window once the operation is completely done. Hence handling more than one window in the application.

Multiple Window Handling in Selenium

Conclusion:

- We defined the types of alert and shown them with a screen shot.
- Demonstrated handling the Alert with Selenium WebDriver using particular scenario.
- Handled multiple windows with Selenium WebDriver using particular scenario.