**In our project , I faced a challenge as unable to click on a web element, that is overlapped by another web Element and we got an issue as element click intercepted exception issue.**

**And I tried different options as**

* **Driver.findElement(By.id(“”)). click();**
* **Also, I tried using actions class**
* **and also tried using JavaScript executor**
* **and I also tried keys dot enter or keys dot Return in**

**driver. Find Element by id .send Keys**

**Still, Web Element click operation is working randomly.**

**So, We had created a web element object reference and**

**scrolled at that specific web element by using JavaScript executor and I had used explicit wait condition as until element to be clickable condition,**

**like that we had fixed that issue by writing code in try, catch exception block of Element Click Intercepted Exception .**

**Like that we fixed that kind of challenge**

**So , Web element click operation is working fine all the times perfectly .**

**And another challenge as**

In our application, we have to execute scripts on Edge browser and Chrome browser combinations.

Scripts executed perfectly in Chrome browser, but the script execution failed in Edge browser . Because of zoom level settings defaulted to 150 %.

We fixed this issue by using java script executor and fixed browser zoom level setting back to 100 % .

**We had written script like**

**//We have to create JavascriptExecutor Reference Object**

**JavascriptExecutor js = (JavascriptExecutor)driver;**

**//and to set 100 % zoom setting**

**String zoom\_setting="document.body.style.zoom='1.0'";**

**js.executeScript(zoom\_setting);**

Like That we fixed browser zoom level settings to 100 % and scripts got executed perfectly in edge browser as well

**And another challenge as**

In our application Web element properties like id ,name, text are changing dynamically every time in the web pages .

To handle these kind of dynamic web elements, we had used different Xpath access techniques.

As

* **Starts with the text of.**

Example : //tagName[starts-with(@id,’fixed description of id ’)]

Example : //tagName[starts-with(text(),’fixed description text’)]

* **Contains text of**.

Example : //tagName[contains(@id,’fixed description of id ’)]

Example : //tagName[contains(text(),’fixed description text’)]

* **Following**
* **Following siblings**
* **Preceding**
* **Preceding siblings**
* **Ancestor**
* **Child**
* **Parent**

**Like that we handled dynamic web elements in selenium**

**And another challenge as**

In our application frames are loading very slowly on the web page

To fix this issue , we had used explicit wait condition as

Wait<WebDriver> wait = **new** WebDriverWait(*driver*, Duration.*ofSeconds*(15));

wait.until(ExpectedConditions.*frameToBeAvailableAndSwitchToIt*("frameName"));

*driver*.switchTo().frame("frameName");

**And another challenge as**

**We have to download the files from the web portal**

**But Selenium could not able handle Windows based popups and prompts to down load the files from desktop interactions ,**

**To handle this issue ,we had used java.awt.robot class**

**, String Selection class and toolkit class files from java.awt package.**

**We are able to populate text in windows based popups**

**Able to invoke CTRL + C copy action , CTRL+V paste action**

**And able to press ENTER key button usng by using robot class methods**

We had Created object of Robot class

Robot robot = **new** Robot();

// we Pressed CTRL+V paste action by writing code as

robot.keyPress(KeyEvent.***VK\_CONTROL***);

robot.keyPress(KeyEvent.***VK\_V***);

robot.keyRelease(KeyEvent.***VK\_CONTROL***);

robot.keyRelease(KeyEvent.***VK\_V***);

// we Pressed Enter key button by writing code as

robot.keyPress(KeyEvent.***VK\_ENTER***);

robot.keyRelease(KeyEvent.***VK\_ENTER***);

// and To populate text in windows based prompt, we had written code as

**StringSelection sel = new StringSelection(filepath);**

**Toolkit.*getDefaultToolkit*().getSystemClipboard().setContents(sel,null);**

**Like that we are able to upload files using java.awt.robot class in java**

**And another challenge as**

**We have to enter capcha text in the text field.**

**But, by using selenium , we cannot able to extract capcha text directly ,**

**so w had used 3rd party plugin maven dependency as com.asprise.ocr .**

**and we had imported supporting class files as**

****

**And To enter the capcha text**

**First we have to get capcha image url**

**and we had written code as**

****

**And next we had written code as**

****

****

**Driver.findElement(By.id(“”)).sendKeys(capchatext);**

**Like that we passed capcha text in sendKeys**

Like that we are able to handle capcha by using 3rd party plugin as

**com.asprise.ocr package**