1. About Framework  
   **2. Collections  
   3. Vector and Array list difference**4. 3 ok buttons with same id and other attributes, how you click them  
   5. You have to test a module without any information, what would be the approach  
   6. Suppose you have 2 builds in a day, how you can make sure that application is stable enough.  
   7. What are the benefits of TestNG   
   8. What is Page Object Model  
   9. Method overloading and overriding  
   **10. What is the most common exception in Selenium**11. Test Scenario  
   12. Suppose you are testing a login functionality, you have to test negative scenario via automation, how to do this?  
   **13. How to automate cascading drop downs.**14. You have saved some information, how to make sure that it gets saved with the help of Automation  
   15. API testing  
   **16. How to skip tests in TestNG**

…………………………………………………………………………………………………………………………………………………….

1. In which project you used selenium
2. What challenges you face duration of project
3. How much experience do you have in selenium
4. How you handle exception handling
5. What framework you used and explain.
6. Have you used jmeter for performance and load testing
7. **How you will handle dynamic xpath in selenium**
8. Why selenium is better than other tool

1a – e Project = JINA(GITA), OSSI and ecommerce admin section

2a- Image **reorganization** and solved with sikuli tool,

3a- 1.2 year

4a- Hybrid= Data driven + Page Object Model

TestNG + Extent report+Maven+Git+Jenkin

5a –

8a – cost effective, support multiple OS/browsers/language(java, php, python,c#)

………………………………………………………………………………………………………………………….

**Part 2- Create different page library in page factory**

**Part 3 – Create different Tastcase**

**Created data driven approach**

**@CacheLookup:**

One issue it gives when page get refresh or DocumentObjectModel property changes then stale element exception will display so @CacheLookup will be use only when we are sure that field element wouldn’t be change that case only we will be use @CacheLookup. Using this script will be fast

**WebDriver Fire Event:** use to generate selenium action logs

**Throwable** : Handle exceptions as well as errors

Extent Report – 2.41.2 latest version – 3.0

Extent report listener we have to write and it listen each and every test case execution activity

**Sample site for creating script:**

https://freecrm.co.in/

**Log4j API** to generate logs

@DataProvider

@FindBy

**Need to learn:”**

Maven Architecture

<https://www.youtube.com/watch?v=t5hoD4D0Jo0>

Maven: Build automation tool 3.3.9 version

**Log4j API**

<https://www.youtube.com/watch?v=C7TEuhgVDYo>

version-1.2.17

**Different labels of Log:**

Info | warning | error | fatel

Latest:

<https://www.youtube.com/watch?v=C7TEuhgVDYo>

Even java developer also used this api for generate the logs in application server(tomcat)

Questions:

1. get and navigate difference

get- wait all the web element of page is loaded then only we can perform any event

navigate- back, forward and refresh methods

get(url)

driver.navigate().to(url)

1. quit and close difference

close- close the browser which is in focus

quit- all the browser will close and web **driver** **instance** itself will also close

1. implicit wait: Beginning of test---5 secs --- 3 secs

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);

1. implicit and explicit difference:

explicit targets the specific web element 120 sec

1. how many ways we can handle frames in application using web driver methods

driver.switchTo().frame(id)-id, name, web element

driver.switchTo.frame(id, name, web element)

1. handle 3rd child window

Set<String> s=driver.**getWindowHandles**();

it=s.iterator(); //0th index of set box

it.next();

String windowid3rd=it.next();

**driver.switchTo().window**(windowid3rd)

This is based on java Collection

1. **handle https certification**

we need to set capability

DesiredCapabilities cap = DesiredCapabilities.chrome();

Set ACCEPT\_SSL\_CERTS variable to true

cap.setCapability(CapabilityType.ACCEPT\_SSL\_CERTS,true)

DesiredCapabilities cap = DesiredCapabilities.chrome();

cap.setCapabilities(CapabilityType.ACCEPT\_SSL\_CERTS,true);

1. Different types of locators in selenium webdriver

Xpath, id, name, css, type, tagname, classname, linkText

Xpath,id,name,css,type

By.name()

By.id()

By.linkText()

By.PartialLinkText()

By.className()

By.tagName(a / div / ) //can check broken links driver.findElements(By.tagName(div))

By.cssSelector() with Tag and Attribute: driver.findElement(By.cssSelector(“input[name=’phone’]”))

Tag and Id:

driver.findElement(By.cssSelector(“input# id name ”))

1. Xpath and css syntax with id and tag

Xpath= //tagname[@id=’value’]

Css=tagname[id=value]

Contains:

Driver.findElements(By.xpath(“//input[contains(@class,’form control’)]”))

1. Contains regular expression in xpath

Id=uo12, uo34, uo63

Xpath= //tagname[contains(@id=’uo’)]-----------this complete thing is locator

Css = tagname[id\*=’uo’]

List<WebElement> elements = driver.findElements(By.name("name"));

List <WebElement> elements =driver.findElements(By.name(“name”));

System.out.println("Radio button text:" + elements.get(i).getAttribute("value"));

System.out.println(“Radio button text:”, +elements.get(i).getAttribute(“value”));

1. Class for handle dropdown in webdriver

**Select**

driver.findElement(By.cssSelector("input#Email")).sendKeys("Software Testing Material");

driver.findElement(By.cssSelector(“input#Email”)).sendKeys(“gfhgh”);

1. Method to handle if checkbox is selected

Driver.findElement(“locator”).isSelected()

1. Validate element if visible or in hidden mode

Driver.findElement(“locator”).isDisplayed()

1. Count similar object list in web page

driver.findElements(By.className(“abc”)).size()

driver.findElements(by.className(“sfd”)).size();

1. Importance of desiredcapabilities: it is a class which describe your browser,os and it’s properties, how browser react

With Selenium Grid- Distribute the tests across multiple machines that can be multiple OS.

So with the help of desired capabilities

1. Enter text in caps loack

Driver.findElement(By.xpath(“dsf”)).sendKeys(keys.SHIFT,”hello”);

1. **Mouse over** on page

**Actions** a = new Actions(driver);

a.moveToElement(locator).build().perform();

1. Handle java alert

Driver.switchToAlert();

Driver.**switchToAlert**();

1. Count total link in page

Driver.findElements(By.tagName(“a”)).size();

findElements(by.tagName(“a”)).size();

1. How to validate if we are navigated to child window successfully

Driver.getTitle();

1. Relative and Absolute xpath

Absolute: start from node—html/body -- navigates to actual element

1. Sample xpath syntax to handle parent from child object

//parent/child

1. What driver is must to run tests in Firefox browser

Gechodriver- from **version 43** we have to use gechodriver to run our test

1. What driver is must to run tests in Chrome browser

Chrome driver

1. Set driver in your systems

System.setProperty(“webdriver.chrome.driver”, “location”)

System.setProperty(“webdriver.gechodriver.driver”, “location”)

System.setProperty(“webdriver.gechodriver.driver”,”location”);

1. findElement and findElements difference

findElements :similar behavior from multiple object

1. list out two methods available in explicit wait

common methods: visibilityOfElementLocated and presenceOfElementLocated

visibilityOfElementLocated: wait till the element is visible

WebDriver driver;

WebDriver driver;

driver = new ChromeDriver();

driver = new ChromeDriver();

WebDriverWait wait=new WebDriverWait(driver, 20);

WebDriverWait wait= new WebDriverWait(driver,20);

WebElement we = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xPath()))

we.click();

1. take screenshots with selenium webdriver

File src= (TakesScreenshot) driver.getScreenshotsAs(OutputType.FILE);

Try{ FileUtils.copyFile(src,new File(“D://selenium /error.png”) }

catch(IOException e) { System.out.println(e.getMessage()); }

1. hit enter command

Driver.findElement(By(locator).sendKeys(Keys.ENTER);

NoAlertPresentException();

NoSuchWindowException();

NoSuchElementException();

NoSuchFrameException();

WebDriverException();

TimeoutException();

1308/77755/00467

08/04/2017 10:54:51

**POM step1:**

POM – Page Object Model- it is a approach-design pattern..

Using POM with DD approach and Hybrid framework

POM+Data Driven approach+TestNG

Login page

Home page

Registration page

Search page

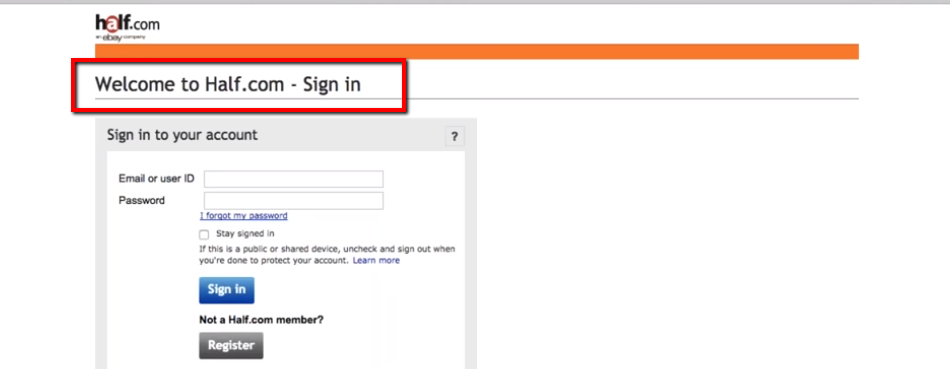
Add to cart page

Checkout page

Payment page

**viewdemo**

**Header example:**



In POM we have to create separate java class for each page

Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can **manage a project's build**, **reporting** and documentation from a **central piece of information**.

POM for create a object repository and Page Factory for maintain object repository