1. Mailing -Inbox scenario
2. Verify and check whether if the new mail is coming then notification is showing or not when we open inbox that mail is present or not.
3. Verify and check After mailing whether mail id is correct or not.
4. Verify and check whether inbox page is opening or not.
5. Verify and check whether date of the mail is correct or not.
6. Verify and check whether time is correct or not.
7. Verify and check whether sender mail id is present or not in inbox.
8. Verify and check whether sender name is present or not in inbox.
9. Verify and check whether when you click on inbox and click on new mail it is opening that page or not.
10. Then Write the text message whatever you want to send. and send message with a valid credentials.
11. Login and logout scenario’s
12. Verify and check whether on enter valid email id and invalid password it is login or not properly
13. Verify and check whether on putting email id blank and enter valid password it is login or not.
14. Verify and check whether on enter valid email id and putting on blank password it is login or not.
15. Verify and check whether logout button is present or not in the home page.
16. Verify and check whether click on logout button login page is displayed or not.

3). Ans.

a) The highest priority issue in portal/site when a very basic and critical features of the application is not working properly which is affected other code and work of application.

Example:- suppose that a user is not able to login to the application by enter a valid username and password. Login the application is very important feature of the application which is may be affect other code and work features in the Application because we can not able to do further activities without login in the application we can not do anything without loing in the application.

1. Scenario’s for Payment on the Order placed on the Flipkart.
2. Verify and check after searching the product wether product is correct or not.
3. Verify and check whether after adding product to the Add to cart and confirm the order placed , then payment option is coming or not.
4. Verify and whether after selecting for payment confirmation message is coming or not.
5. Verify and whether while selecting payment ,different payment options is coming or not like ebanking. Debit card, credit Card, Emi etc.
6. Verify and check whether cash on delievery option is present or not .
7. Verify and check whether after entering bank details , pay button is clicking or not.
8. Verify and check whether it is able to select my bank options for payment is coming or not.
9. Verify and whether after doing payment, it is bank options is coming to bank page or not.
10. Verify and check whether after doing payment, notification is coming or not for successful payment.
11. Smoke testing.
12. Testing the basic and critical features of the application is called smoke testing . we do smoke testing to verify whether the build is stable or not after testing its basic and critical featues of an applications.
13. Checking basic and critical feature of the application for every positive secanario’s.
14. Whenever new build we get from development team we check for basic and critical feature of the application for every positive scenario’s to verify the build early stage to find any blocker bog.

Advantage

a)Test engineer get enof time to test the application.

b) developer is not sitting idealy they get enof time to fix t5he bug.

c) test execution cycle time not get affected.

Core java Questions :

1. Class sample {

Public static void main(string[]args)

{

for(int i=0; I <=5; i++)

{

For(int j=5; j>=0; J--)

{

System.out.println(J);

}

}

1. C lass CharactersampleTest {

Public static void main(String[] args)

{

String str=”JAVA”;

String[]arr1=str1.splite(“ ”);

System.out.println(“string is ”:+str1);

3 class StringReverse

{

Public static void main(String[] args)

String str=” Test Data”;

System.out.println(str);

System.out.println(“printing in reverse order”:+str.reverse());

}

}

1. class sample

{

Public static void main(String[] args)

{

Int Array[] = { 9,3,6,1,7 };

For( integer arr: i)

{

System.out.primntln(arr);

5. class StringPallidrome

{

Public static void main(String[] args)

{

String s= “bob”;

String org\_string= s;

String rev = “ “;

For(int i=s.length-1; i>=0; i--)

{

Rev= rev + s.charAt[i]

}

System.out.println(rev);

}

If(rev.equals( org\_string))

{

System.out.println(“String is palindrome”)

}

}

Selenium Questions :

1)

Class MultiWindowTest

{

Public static void main(String[]args)

{

WebDriver driver= new FireFoxDriver();

//navigate to flipkart

driver.get(“//http:flipkart.com”);

//close popup

driver.FindElenent(By.xpath(“//input[text()=’x’]”)).click();

driver.FindElement(By.Name(‘”q”)).sendkeys(“Apple”,Keys.Enter);

webElement wb1= FindElenent(By.xpath(“//input[text()=’Apple 32gb’]”);

webElement wb2= FindElenent(By.xpath(“//input[text()=’Apple 64gb’]”);

//capture the session id

List<webElement> setLst= Driver.getWindowHandles();

Iterator(String) itr=setlst.iterator();

String childSessioniD=itr.next();

String ParentSessionID= itr.next();

driver.switchTo.Window(childSessionID);

driver.FindElements(By.Xpath(“//input[text()=’buy’])).click();

driver.close();

}

}

2)

a) implicit wait:

implicit wait always wait for entire html Documents page to be loaded in gui before performing action on the elements.

Technically implicit wait monitor entire html documents to be loaded in gui before forming action on the elements. If the elements is available on the before maximum time limit then driver control release next line insead of waiting maximum time limit. If the anyboady element is not available before maximum time limit then throws TimeOutException. And stop entire execution

Disadvantage:

implicit wait can not use in ajex appli8cation because Ajex application contain lot of dynamic elements.

b) Explicit wait:

Explicit wait always wait for expected elements to available in gui.

Technically Explicit wait monitor html documents to be loaded in gui . Explicit wait always wait for expected elements to available in gui. And it will every 500 milli second

If the elements is available on the before maximum time limit then driver control release next line insead of waiting maximum time limit. If the anyboady element is not available before maximum time limit then throws TimeOutException. And stop entire execution.

Advantage:

Explicit Wait can be use in ajex applicatiin because ajex application contain lot of dynamic eelements’

c)fluent wait:

fluent wait is a extended version of Explicit Wait we can customize dealy time and searchinh time. It means Fluient wait always wait for expected element to available in gui bu we can customize dealy timing and searching timing like miniutes, second.

1. Mouse movement countroler:

Class DragAndDropTest

{

Public static void main(String[]args)

{

WebDriver driver= new FireFoxDriver();

//navigate to flipkart

driver.get(“//http:dragAndDrop”);

webelement src= driver.findElements(By.id(“Draggable”));

webelement dest= driver.findElements(By.id(“Dropable”));

Actions act= new Actions(driver);

act.dragAndDrop(src, dest).build().perfum();

}

}

4}Screenshot:

Class ScreenShot

{

Public static void main(String[]args)

{

WebDriver driver= new FireFoxDriver();

//navigate to flipkrt

driver.get(“https:flipkart.com”);

EventFiringWebDreiver edriver = new EventFiringWebDreiver(driver);

File src= edriver.geatScreenShotAs(FileType.output);

File dest= edriver.setfileType(“//c:/pintu/test.screenshot”);

utilsFile.(src,dest);

}

}

5}

Class Sample

{

Public static void main(String[]args)

{

WebDriver driver= new FireFoxDriver();

//navigate to flipkart

driver.get(“https:flipkart.com”);

webElements wb =driver.findElements(By.LinkText(“//a));

for (webElements wb: elements)

{

System.out.println(wb.gettext());

}

}