|  |
| --- |
| CGI proovitöö testiplaan/Raport/Java  Merle orgmets  Email: merle.orgmets@gmail.com |

## INTRODUCTION

The purpose of this Software Test Plan (STP) is to develop Automation test plan and Automation Test steps to assess eleven different functionalities of the http://automationpractice.com/index.php CGI Eesti Company.

**USED DEVICES AND OP SYSTEMS IN AUTOMATION TESTING**

Device: Laptop

Op System: Windows 10 Home

64-bit operating system

**USED WEBSITE FOR TESTING**

Google Chrome

Version 92.0.4515.159 (Official Build) (64-bit)

**USED TESTLANGUAGE FOR AUTOMATION**

JAVA

**TEST PLAN**

The automationpractice website will consist of eleven basic areas of testing including one negative testcase. The first item of testing includes validating the Browser. The second item assessed is how the information is stored within the browser website. The third item under test is how the information will be retrieved and displayed to the user based on their initial request.

**Scope**

The scope of this document is to test the automationpractice website system by each component, component integration, and system functionality. However, due to the lack of available project time, this STD will only include seven simple Automation testing procedures and one simple negative testcase procedure.

**Resource Material**

Gmail document

Portable framework Selenium for testing web applications.

Robot framework

Eclipse for developing Java applications

**Automation testing**

This process involves the Automation testing of particular system components. Tested for their input, output, and module procedures used Java programming language.

**TEST DESIGN**

* 1. **Approach**

**Open Browser/Navigate to browser automationpractice.com**

Test opens Google chrome and <http://automationpractice.com/> webpage opens.

*Expected behavior: Value = True ( Test Passed )*

**Maximize Browser window**

Browser zoom into maximum size

*Expected behavior: Value = True ( Test passed )*

**Choose TAB Women**

When the user selects Category women , the subcategory with selections opens

E*xpected behavior: value = True ( test Passed )*

**Choose TAB TOPS**

User selects tab tops and subcategory with selections opens

*Expected behavior: value = true ( test passed )*

**Select selection T-SHIRTS**

User navigates to the selection t-shirts and different selections opens

*Expected behavior: value = true ( Test passed )*

**Go to the main page, using webpage logo**

User clicks main page logo and navigates in to the main page. Website main page opens.

*Expected behavior: value = true ( test passed )*

**Select Sign In Button**

user clicks log in button. website with log in credentials opens

*Expected behavior: value = true ( test passed )*

**Navigate to Username and Password section/ Insert wrong e-mail address**

User adds wrong e-mail address into e-mail field.

*Expected Behavior: Value = True ( Test Passed )*

**Select Sign In button**

User clicks Sign in button, after the wrong e-mail address is inserted.

*Expected Behavior: Value = True ( Test Passed*

**Check is the error message correct**

Test checks is the given message error correct and if it is then test is passed.

*Expected Behavior: Value = True ( Test Passed )*

**Close browser**

Browser closes

*Expected behavior: value = true ( Test Passed )*

**Used Test Code Java**

1. **Open Browser/Navigate to browser automationpractice.com**

WebDriver driver = **new** ChromeDriver();

//Navigate to a website

driver.get("http://automationpractice.com/index.php");

**2.** **Maximize Browser window**

//Mazimize browser window

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

**3.** **Choose TAB Women**

// Choose TAB Women

WebElement svgObject = driver.findElement(By.*xpath*("//\*[@id='block\_top\_menu']/ul/li[1]/a"));

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

builder.click(svgObject).build().perform();

Thread.*sleep*(4000);

**4.** **Choose TAB TOPS**

//Choose TAB TOPS

WebElement svgObject2 = driver.findElement(By.*xpath*("//\*[@id=\"categories\_block\_left\"]/div/ul/li[1]/a[contains(text(),'Tops')]"));

Actions builder2 = **new** Actions(driver);

builder2.click(svgObject2).build().perform();

**5.** **Select selection T-SHIRTS**

//Choose TAB T-shirts

**boolean** staleElement = **true**;

**while**(staleElement){

**try**{

driver.findElement(By.*xpath*("//\*[@id=\"subcategories\"]/ul/li[1]/div[1]/a/img")).click();

staleElement = **false**;

} **catch**(StaleElementReferenceException e){

staleElement = **true**;

}

}

**6.** **Go to the main page, using webpage logo**

//Go to the main page using page logo ( Your Logo a new experience )

driver.findElement(By.*xpath*("//\*[@class='logo img-responsive']")).click();

**7.** **Select Log In Button**

// Push Log in button

driver.findElement(By.*xpath*("//\*[@id=\"header\"]/div[2]/div/div/nav/div[1]/a")).click();

**8.** **Navigate to Username and Password section/Insert wrong e-mail address**

//Insert wrong e-mail aadress

driver.findElement(By.*xpath*("//input[@id='email']")).sendKeys("Helloo");

**9.** **Select Sign In button**

//Select Sign in buttondriver.findElement(By.*xpath*("//\*[@id=\"SubmitLogin\"]/span")).click();

Thread.*sleep*(2000);

**10.** **Check is the error message correct**

//Check is the error message correct

String actual\_error= driver.findElement(By.*xpath*("//\*[@id=\"center\_column\"]/div[1]/ol/li")).getText();

String expected\_error="Invalid email address.";

Assert.*assertEquals*(actual\_error,expected\_error);

System.***out***.println("Test completed");

**11. Close browser**

// Close browser

driver.close();

Antud automaattesti kirjutamise ning jooksutamise jaoks valisin testdata editori Eclipse. Miks seda just kasutasin? Otsustasin kirjutada testikoodi java programmeerimiskeelt kasutades. Eclipse valisin selle tõttu, et antud programmist on võimalik erinevaid raamistikke kasutada.

**Näide Eclipse-st:**



