



**Department of Computer Science and Engineering**  
**(IoT, Cybersecurity with Blockchain Technology)**  
**B.Tech. Sem: VII Subject: Software Testing & Quality Assurance**

**Experiment 5**

**Name: Hiren Darji**

**SAP ID: 60019230114**

**Aim: - Writing Test Scripts for the test cases.**

**CU1 – Verify Name Field: Name cannot be empty**

```
import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint
import static com.kms.katalon.core.testcase.TestCaseFactory.findTestCase
import static com.kms.katalon.core.testdata.TestDataFactory.findTestData
import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject
import static com.kms.katalon.core.testobject.ObjectRepository.findWindowsObject
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI
import com.kms.katalon.core.util.KeywordUtil
import com.kms.katalon.core.webui.driver.DriverFactory
import org.openqa.selenium.By
import org.openqa.selenium.WebDriver
import org.openqa.selenium.WebElement
import org.openqa.selenium.support.ui.WebDriverWait

import org.openqa.selenium.support.ui.ExpectedConditions

// ----- Common Setup -----
WebUI.openBrowser('')
WebUI.maximizeWindow()

WebDriver driver = DriverFactory.getWebDriver()

driver.get('https://code-collab-sage.vercel.app/')

// ----- Wait and Click on Sign Up -----
// NOTE: use older WebDriverWait constructor (driver, timeoutInSeconds)
WebDriverWait wait = new WebDriverWait(driver, 10) // <-- fixed constructor

// prefer button selector, fallback to absolute xpath if needed
By signUpBtnBy = By.xpath("//button[normalize-space()='Sign Up']")

WebElement signUpBtn =
wait.until(ExpectedConditions.elementToBeClickable(signUpBtnBy))
signUpBtn.click()

// ----- CU1: Name Field Cannot be Empty -----
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("display-name")))

WebElement displayNameElem = driver.findElement(By.id("display-name"))
```



```
WebElement emailElem = driver.findElement(By.id("email-address"))

// ensure input is cleared and blank
displayNameElem.clear()
displayNameElem.sendKeys("")

// shift focus to email field to trigger HTML5 validation
emailElem.click()

// capture browser validation message
String validationMsg = displayNameElem.getAttribute("validationMessage")
KeywordUtil.LogInfo("Validation message captured: " + validationMsg)

if (validationMsg != null && validationMsg.toLowerCase().contains("fill out this field")) {
    KeywordUtil.markPassed("CU1 Passed: Error message displayed for empty name field")
    WebUI.takeScreenshotAsCheckpoint("CU1_Passed")
} else {
    WebUI.takeScreenshotAsCheckpoint("CU1_Failed")
    KeywordUtil.markFailedAndStop("CU1 Failed: Error message not displayed for empty name")
}

// ----- Teardown -----
try {
    driver.quit()
} catch (Exception ignored) { }
```



## CU2 – Name cannot be Numeric

```
import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject

import com.kms.katalon.core.util.KeywordUtil
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI
import com.kms.katalon.core.webui.driver.DriverFactory

import org.openqa.selenium.By
import org.openqa.selenium.WebDriver
import org.openqa.selenium.WebElement
import org.openqa.selenium.Keys
import org.openqa.selenium.support.ui.ExpectedConditions
import org.openqa.selenium.support.ui.WebDriverWait

// ----- Test data (as requested) -----
String testName = '1212'
String testEmail = 'pharosa@gmail.com'
String testPassword = 'abcd45@123'

// ----- Setup -----
WebUI.openBrowser('')
WebUI.maximizeWindow()

WebDriver driver = DriverFactory.getWebDriver()
driver.get('https://code-collab-sage.vercel.app/')

// ----- Wait and Click on Sign Up -----
WebDriverWait wait = new WebDriverWait(driver, 15) // 15s timeout for slower renders
By signUpBtnBy = By.xpath('//button[normalize-space()="Sign Up"]')
WebElement signUpBtn =
wait.until(ExpectedConditions.elementToBeClickable(signUpBtnBy))
signUpBtn.click()

// ----- Wait for the sign-up form fields -----
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id('display-name'))))
WebElement displayNameElem = driver.findElement(By.id('display-name'))

// Fill name (numeric), email, password
displayNameElem.clear()
displayNameElem.sendKeys(testName)

// Some forms may render email/password with different ids; use common ones
// Try common ids for email and password; adapt if your app uses different attributes
By emailBy = By.id('email') // preferred id
By passwordBy = By.id('password') // preferred id

WebElement emailElem = null
WebElement passwordElem = null

// Try to locate email element - if not found by id, try input[type='email']
try {
    emailElem = driver.findElement(emailBy)
} catch (Exception ignored) {
    try {
```



```
emailElem = driver.findElement(By.xpath("//input[@type='email' or
contains(@name,'email')]"))
    } catch (Exception ignored2) {
        emailElem = null
    }
}

try {
    passwordElem = driver.findElement(passwordBy)
} catch (Exception ignored) {
    try {
        passwordElem = driver.findElement(By.xpath("//input[@type='password' or
contains(@name,'password')]"))
    } catch (Exception ignored2) {
        passwordElem = null
    }
}

// If fields found, populate them
if (emailElem != null) {
    emailElem.clear()
    emailElem.sendKeys(testEmail)
} else {
    KeywordUtil.LogInfo("Email field not found by usual locators; continuing
without filling email (check locators).")
}

if (passwordElem != null) {
    passwordElem.clear()
    passwordElem.sendKeys(testPassword)
} else {
    KeywordUtil.LogInfo("Password field not found by usual locators; continuing
without filling password (check locators).")
}

// Trigger blur to make validation run (ensure validation is triggered)
displayNameElem.sendKeys(Keys.TAB)

// Small short wait for frontend processing (explicit waits follow for the error
element)
Thread.sleep(500) // half-second pause to let client-side validation start

// ----- Submit the form -----
// Try to find the form submit button inside the visible modal/form
WebElement submitBtn = null
try {
    // prefer a button[type='submit'] inside form/modal
    submitBtn = driver.findElement(By.xpath("//form//button[@type='submit' and
(normalize-space()!= ' ')] [1]"))
} catch (Exception e1) {
    try {
        // fallback: clickable button with text like 'Sign up', 'Create account',
'Submit'
        submitBtn =
driver.findElement(By.xpath("//button[contains(translate(normalize-
space(.), 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxyz'), 'sign up') or
contains(translate(normalize-
space(.), 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxyz'), 'create
```



```
account') or contains(translate(normalize-
space(.), 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxyz'), 'submit'))][1]"
))
    } catch (Exception e2) {
        submitBtn = null
    }
}

if (submitBtn != null) {
    try {
        wait.until(ExpectedConditions.elementToBeClickable(submitBtn))
        submitBtn.click()
    } catch (Exception clickEx) {
        KeywordUtil.LogInfo("Submit button found but click failed: " +
clickEx.getMessage())
        // attempt JS click as fallback
        try {
            ((org.openqa.selenium.JavascriptExecutor)
driver).executeScript("arguments[0].click();", submitBtn)
        } catch (Exception jsEx) {
            KeywordUtil.LogInfo("JS click fallback also failed: " +
jsEx.getMessage())
        }
    }
} else {
    KeywordUtil.LogInfo("Submit button not found with fallback locators. Form may
auto-validate without explicit submit.")
}

// ----- Wait for the error message "numbers are not allowed" -----
// Use class-based XPath (robust) + normalize-space + translate for case-
insensitive exact text match
By numericErrorXpath = By.xpath(
    "//div[contains(@class, 'text-red-600') or contains(@class, 'bg-red-50')]" +
    "[contains(translate(normalize-
space(.), 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxyz'), 'numbers are
not allowed')]"
)

boolean cu2_flag = false
try {
    // wait up to 15 seconds for the error to appear (react can be async)
    WebDriverWait waitError = new WebDriverWait(driver, 15)
    WebElement errorElem =
waitError.until(ExpectedConditions.visibilityOfElementLocated(numericErrorXpath))
    cu2_flag = errorElem != null && errorElem.isDisplayed()
} catch (Exception e) {
    cu2_flag = false
    KeywordUtil.LogInfo("Error element not found or not visible: " +
e.getMessage())
}

// ----- Result handling -----
if (cu2_flag) {
    KeywordUtil.markPassed('CU2 Passed: Error message displayed for numeric name
("numbers are not allowed").')
    WebUI.takeScreenshot()
}
```



```
} else {  
    WebUI.takeScreenshot()  
    KeywordUtil.markFailedAndStop('CU2 Failed: Numeric name error not displayed or  
XPath mismatch')  
}  
  
// ----- Teardown -----  
try {  
    driver.quit()  
} catch (Exception ignored) {}  
  
WebUI.closeBrowser()
```



### CU3 – Name cannot have Special Characters

```
import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject

import com.kms.katalon.core.util.KeywordUtil
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI
import com.kms.katalon.core.webui.driver.DriverFactory

import org.openqa.selenium.By
import org.openqa.selenium.WebDriver
import org.openqa.selenium.WebElement
import org.openqa.selenium.Keys
import org.openqa.selenium.support.ui.ExpectedConditions
import org.openqa.selenium.support.ui.WebDriverWait

// ----- Test data -----
String testName = 'Hiren@' // Name with special character
String testEmail = 'pharmosa@gmail.com'
String testPassword = 'viral123@'

// ----- Setup -----
WebUI.openBrowser('')
WebUI.maximizeWindow()

WebDriver driver = DriverFactory.getWebDriver()
driver.get('https://code-collab-sage.vercel.app/')

// ----- Click on Sign Up -----
WebDriverWait wait = new WebDriverWait(driver, 15)
By signUpBtnBy = By.xpath('//button[normalize-space()="Sign Up"]')
WebElement signUpBtn =
wait.until(ExpectedConditions.elementToBeClickable(signUpBtnBy))
signUpBtn.click()

// ----- Wait for the form fields -----
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id('display-name'))))
WebElement displayNameElem = driver.findElement(By.id('display-name'))

// Fill name, email, password
displayNameElem.clear()
displayNameElem.sendKeys(testName)

// Locate email and password fields with common locators
WebElement emailElem = null
WebElement passwordElem = null

try {
    emailElem = driver.findElement(By.id('email'))
} catch (Exception ignored) {
    try {
        emailElem = driver.findElement(By.xpath("//input[@type='email' or
contains(@name,'email')]"))
    } catch (Exception ignored2) {
        emailElem = null
    }
}
}
```



```
try {
    passwordElem = driver.findElement(By.id('password'))
} catch (Exception ignored) {
    try {
        passwordElem = driver.findElement(By.xpath("//input[@type='password'
or contains(@name,'password')]"))
    } catch (Exception ignored2) {
        passwordElem = null
    }
}

// Fill email and password
if (emailElem != null) {
    emailElem.clear()
    emailElem.sendKeys(testEmail)
} else {
    KeywordUtil.LogInfo("Email field not found; check locators.")
}

if (passwordElem != null) {
    passwordElem.clear()
    passwordElem.sendKeys(testPassword)
} else {
    KeywordUtil.LogInfo("Password field not found; check locators.")
}

// Trigger blur to fire validation
displayNameElem.sendKeys(Keys.TAB)
Thread.sleep(500) // allow frontend validation to start

// ----- Submit the form -----
WebElement submitBtn = null
try {
    submitBtn = driver.findElement(By.xpath("//form//button[@type='submit' and
(normalize-space()!='')][1]"))
} catch (Exception e1) {
    try {
        submitBtn =
driver.findElement(By.xpath("//button[contains(translate(normalize-
space(.),'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz'),'sign up') or
contains(translate(normalize-
space(.),'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz'),'create
account') or contains(translate(normalize-
space(.),'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz'),'submit')][1]"
))
    } catch (Exception e2) {
        submitBtn = null
    }
}

if (submitBtn != null) {
    try {
        wait.until(ExpectedConditions.elementToBeClickable(submitBtn))
        submitBtn.click()
    } catch (Exception clickEx) {
        KeywordUtil.LogInfo("Submit button found but click failed: " +
clickEx.getMessage())
    }
}
```





```
((org.openqa.selenium.JavascriptExecutor)
driver).executeScript("arguments[0].click();", submitBtn)
    } catch (Exception jsEx) {
        KeywordUtil.LogInfo("JS click fallback failed: " +
jsEx.getMessage())
    }
}
} else {
    KeywordUtil.LogInfo("Submit button not found with fallback locators. Form
may auto-validate.")
}

// ----- Wait for error message "Special characters are not
allowed" -----
By specialCharErrorXpath = By.xpath(
    "//div[contains(@class,'text-red-600') or contains(@class,'bg-red-50')]" +
    "[contains(translate(normalize-
space(.),'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz'),'special
characters are not allowed')]"
)

boolean cu3_flag = false
try {
    WebDriverWait waitError = new WebDriverWait(driver, 15)
    WebElement errorElem =
waitError.until(ExpectedConditions.visibilityOfElementLocated(specialCharErrorXpat
h))
    cu3_flag = errorElem != null && errorElem.isDisplayed()
} catch (Exception e) {
    cu3_flag = false
    KeywordUtil.LogInfo("Error element not found or not visible: " +
e.getMessage())
}

// ----- Result handling -----
if (cu3_flag) {
    KeywordUtil.markPassed('CU3 Passed: Error message displayed for name with
special characters ("Special characters are not allowed").')
    WebUI.takeScreenshot()
} else {
    WebUI.takeScreenshot()
    KeywordUtil.markFailedAndStop('CU3 Failed: Special character name error not
displayed or XPath mismatch')
}

// ----- Teardown -----
try {
    driver.quit()
} catch (Exception ignored) {}

WebUI.closeBrowser()
```



## CU4 Name cannot have first character as blank space

```
import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject

import com.kms.katalon.core.util.KeywordUtil
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI
import com.kms.katalon.core.webui.driver.DriverFactory

import org.openqa.selenium.By
import org.openqa.selenium.WebDriver
import org.openqa.selenium.WebElement
import org.openqa.selenium.Keys
import org.openqa.selenium.support.ui.ExpectedConditions
import org.openqa.selenium.support.ui.WebDriverWait

// ----- Test data -----
String testName = ' Hiren' // First character is a space
String testEmail = 'spacecheck@gmail.com'
String testPassword = 'space123@'

// ----- Setup -----
WebUI.openBrowser('')
WebUI.maximizeWindow()

WebDriver driver = DriverFactory.getWebDriver()
driver.get('https://code-collab-sage.vercel.app/')

// ----- Click on Sign Up -----
WebDriverWait wait = new WebDriverWait(driver, 15)
By signUpBtnBy = By.xpath('//*[@normalize-space()="Sign Up"]')
WebElement signUpBtn =
wait.until(ExpectedConditions.elementToBeClickable(signUpBtnBy))
signUpBtn.click()

// ----- Wait for the form fields -----
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id('display-name'))))
WebElement displayNameElem = driver.findElement(By.id('display-name'))

// Fill name (first character is space), email, password
displayNameElem.clear()
displayNameElem.sendKeys(testName)

// Locate email and password fields
WebElement emailElem = null
WebElement passwordElem = null

try {
    emailElem = driver.findElement(By.id('email'))
} catch (Exception ignored) {
    try {
        emailElem = driver.findElement(By.xpath("//*[@type='email' or
contains(@name,'email')]"))
    } catch (Exception ignored2) {
        emailElem = null
    }
}
```



```
try {
    passwordElem = driver.findElement(By.id('password'))
} catch (Exception ignored) {
    try {
        passwordElem = driver.findElement(By.xpath("//input[@type='password' or
contains(@name,'password')]"))
    } catch (Exception ignored2) {
        passwordElem = null
    }
}

// Fill email and password
if (emailElem != null) {
    emailElem.clear()
    emailElem.sendKeys(testEmail)
} else {
    KeywordUtil.LogInfo("Email field not found; check locators.")
}

if (passwordElem != null) {
    passwordElem.clear()
    passwordElem.sendKeys(testPassword)
} else {
    KeywordUtil.LogInfo("Password field not found; check locators.")
}

// Trigger blur to fire validation
displayNameElem.sendKeys(Keys.TAB)
Thread.sleep(500) // allow frontend validation to start

// ----- Submit the form -----
WebElement submitBtn = null
try {
    submitBtn = driver.findElement(By.xpath("//form//button[@type='submit' and
(normalize-space()!='')][1]"))
} catch (Exception e1) {
    try {
        submitBtn =
driver.findElement(By.xpath("//button[contains(translate(normalize-
space(.),'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz'),'sign up') or
contains(translate(normalize-
space(.),'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz'),'create
account') or contains(translate(normalize-
space(.),'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz'),'submit')][1]"
))
    } catch (Exception e2) {
        submitBtn = null
    }
}

if (submitBtn != null) {
    try {
        wait.until(ExpectedConditions.elementToBeClickable(submitBtn))
        submitBtn.click()
    } catch (Exception clickEx) {
        KeywordUtil.LogInfo("Submit button found but click failed: " +
clickEx.getMessage())
    }
}
```



```
        try {
            ((org.openqa.selenium.JavascriptExecutor)
driver).executeScript("arguments[0].click();", submitBtn)
        } catch (Exception jsEx) {
            KeywordUtil.logInfo("JS click fallback failed: " + jsEx.getMessage())
        }
    }
} else {
    KeywordUtil.logInfo("Submit button not found with fallback locators. Form may
auto-validate.")
}

// ----- Wait for error message "First character cannot be space" -
-----
By firstCharErrorXpath = By.xpath(
    "//div[contains(@class,'text-red-600') or contains(@class,'bg-red-50')]" +
    "[contains(translate(normalize-
space(.),'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz'),'first
character cannot be space')]"
)

boolean cu4_flag = false
try {
    WebDriverWait waitError = new WebDriverWait(driver, 15)
    WebElement errorElem =
waitError.until(ExpectedConditions.visibilityOfElementLocated(firstCharErrorXpath)
)
    cu4_flag = errorElem != null && errorElem.isDisplayed()
} catch (Exception e) {
    cu4_flag = false
    KeywordUtil.logInfo("Error element not found or not visible: " +
e.getMessage())
}

// ----- Result handling -----
if (cu4_flag) {
    KeywordUtil.markPassed('CU4 Passed: Error message displayed for first
character blank space ("First character cannot be space").')
    WebUI.takeScreenshot()
} else {
    WebUI.takeScreenshot()
    KeywordUtil.markFailedAndStop('CU4 Failed: First character blank space error
not displayed or XPath mismatch')
}

// ----- Teardown -----
try {
    driver.quit()
} catch (Exception ignored) {}

WebUI.closeBrowser()
```



## CU10 PASSWORD must have atleast one uppercase and lowercase character

```
import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject

import com.kms.katalon.core.util.KeywordUtil
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI
import com.kms.katalon.core.webui.driver.DriverFactory

import org.openqa.selenium.By
import org.openqa.selenium.WebDriver
import org.openqa.selenium.WebElement
import org.openqa.selenium.Keys
import org.openqa.selenium.support.ui.ExpectedConditions
import org.openqa.selenium.support.ui.WebDriverWait

// ----- Test data -----
String testName = 'Hiren' // Valid name
String testEmail = 'testpassword@gmail.com'
String testPassword = '123456@#' // Invalid password: no uppercase or
lowercase

// ----- Setup -----
WebUI.openBrowser('')
WebUI.maximizeWindow()

WebDriver driver = DriverFactory.getWebDriver()
driver.get('https://code-collab-sage.vercel.app/')

// ----- Click on Sign Up -----
WebDriverWait wait = new WebDriverWait(driver, 15)
By signUpBtnBy = By.xpath('//button[normalize-space()="Sign Up"]')
WebElement signUpBtn =
wait.until(ExpectedConditions.elementToBeClickable(signUpBtnBy))
signUpBtn.click()

// ----- Wait for the form fields -----
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id('display-name'))))
WebElement displayNameElem = driver.findElement(By.id('display-name'))
displayNameElem.clear()
displayNameElem.sendKeys(testName)

// Locate email and password fields
WebElement emailElem = null
WebElement passwordElem = null

try {
    emailElem = driver.findElement(By.id('email'))
} catch (Exception ignored) {
    try {
        emailElem = driver.findElement(By.xpath("//input[@type='email' or
contains(@name,'email')]"))
    } catch (Exception ignored2) {
        emailElem = null
    }
}

try {
```



```
passwordElem = driver.findElement(By.id('password'))
} catch (Exception ignored) {
    try {
        passwordElem = driver.findElement(By.xpath("//input[@type='password' or
contains(@name,'password')]"))
    } catch (Exception ignored2) {
        passwordElem = null
    }
}

// Fill email and password
if (emailElem != null) {
    emailElem.clear()
    emailElem.sendKeys(testEmail)
} else {
    KeywordUtil.LogInfo("Email field not found; check locators.")
}

if (passwordElem != null) {
    passwordElem.clear()
    passwordElem.sendKeys(testPassword)
    passwordElem.sendKeys(Keys.ENTER) // press ENTER to submit
} else {
    KeywordUtil.LogInfo("Password field not found; check locators.")
}

// Trigger blur to ensure frontend validation
displayNameElem.sendKeys(Keys.TAB)
Thread.sleep(500)

// ----- Wait for error message "PASSWORD must have uppercase and
lowercase character" -----
By passwordErrorXpath = By.xpath(
    "//div[contains(@class,'text-red-600') or contains(@class,'bg-red-50')]" +
    "[contains(translate(normalize-
space(.),'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz'),'password must
have uppercase and lowercase character')]"
)

boolean cu10_flag = false
try {
    WebDriverWait waitError = new WebDriverWait(driver, 15)
    WebElement errorElem =
waitError.until(ExpectedConditions.visibilityOfElementLocated(passwordErrorXpath))
    cu10_flag = errorElem != null && errorElem.isDisplayed()
} catch (Exception e) {
    cu10_flag = false
    KeywordUtil.LogInfo("Error element not found or not visible: " +
e.getMessage())
}

// ----- Result handling -----
if (cu10_flag) {
    KeywordUtil.markPassed('CU10 Passed: Password complexity error displayed
("PASSWORD must have uppercase and lowercase character").')
    WebUI.takeScreenshot()
} else {
    WebUI.takeScreenshot()
}
```



```
KeywordUtil.markFailedAndStop('CU10 Failed: Password complexity error not  
displayed or XPath mismatch')  
}
```

```
// ----- Teardown -----  
try {  
    driver.quit()  
} catch (Exception ignored) {}
```

```
WebUI.closeBrowser()
```

### Prepared test cases

