Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică şi Microelectronică

Departamentul Ingineria Software și Automatică

**RAPORT**

Lucrarea de laborator nr. 3

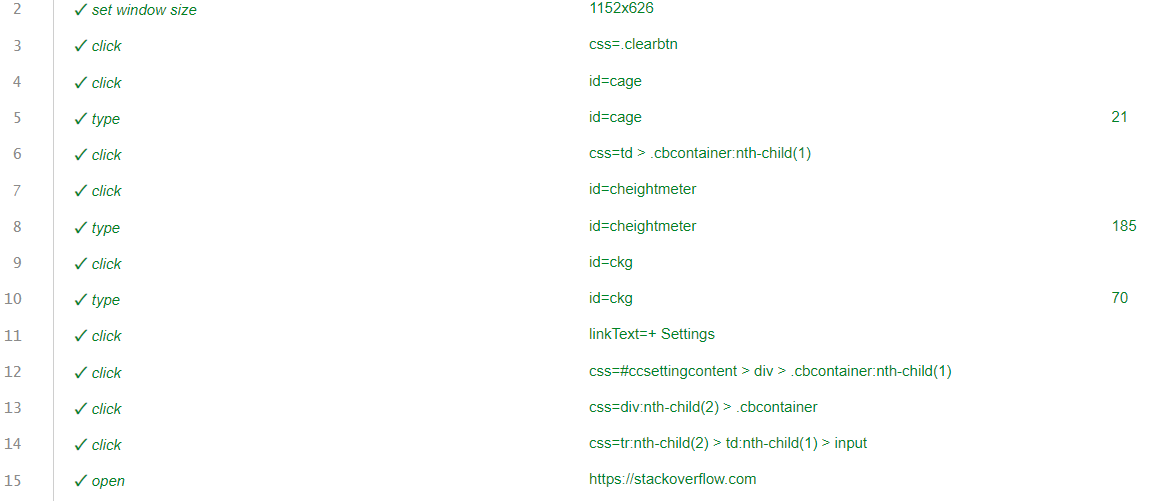
La disciplina „Testare software”

A efectuat: A. Valciuc

A verificat: A. Crîjanovschi

Chișinău – 2023

**Scopul lucrării:** Adăugarea pașilor în test case-uri existente și utilizarea comenzilor în Selenium IDE. Realizarea a două test case-uri pentru login, logout și profile edit.

Figura 1 – Adăugarea unui pas nou

În figura 1 este demonstrată adăugrea unui pas nou în test case-ul creat în cadrul lucrării de laborator nr. 2 și anume pasul 15. După realizarea tututor pașilor are loc deschiderea paginii web alese pentru testarea în cadrul lucrării de laborator curente.

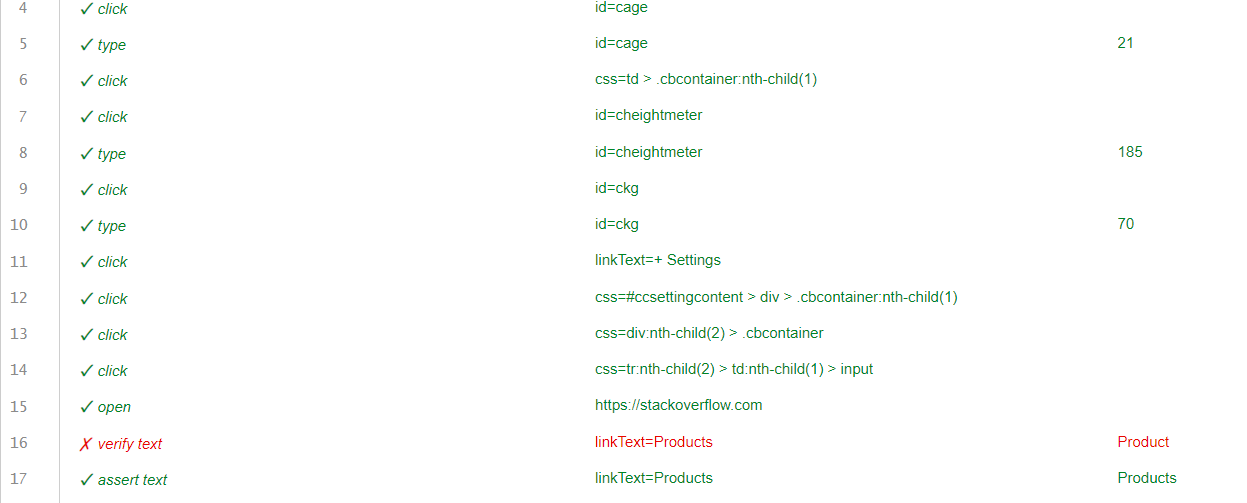


Figura 2 – Adăugarea comenzilor *verify text* și *assert text*

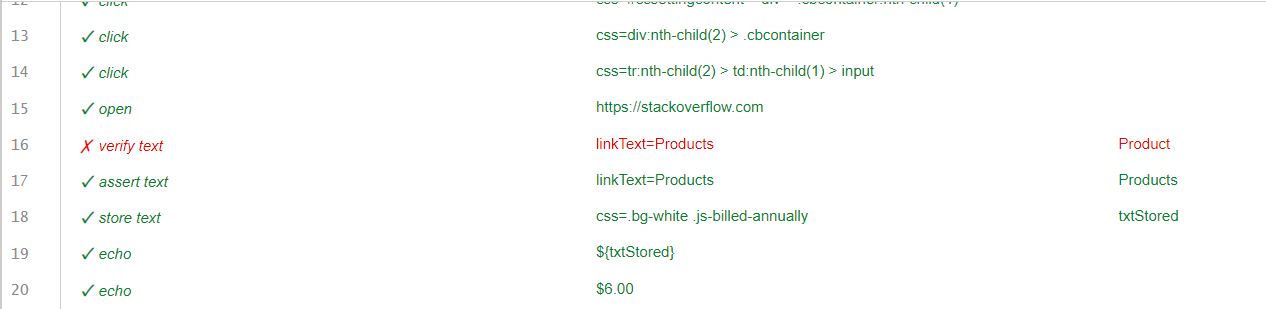
În figura 2 putem observa adăugarea comenzilor verify text și assert text. Prima dintre acestea verifică la coincidență textul introdus spre verificare și textul care este introdus în câmpul de pe website al cărui id este specificat ca target.

Figura 3 – Stocarea textului într-o variabilă

În figura 3 este reprezentată stocarea unui text dintr-un câmp de pe website intr-o variabilă. De asemenea, a fost utilizată comanda echo de două ori urmată atât de numele variabilei, cât și de valoarea numerică introdusă explicit. Rezultatul acestor pași este același, fiind diferit doar modul de utilizare.

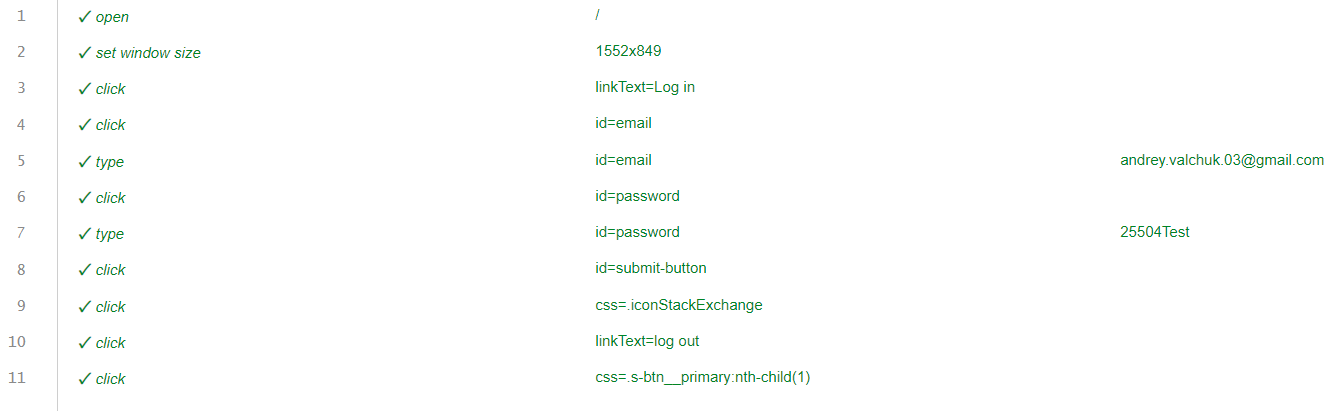


Figura 4 – Elaborarea unui test case pentru *Login* și *Logout*.

În figura 4 este demonstrat test case-ul realizat pentru login și logout pentru website-ul ales. După cum este arătat în figură, toți pașii au fost executați cu succes.



Figura 5 – Elaborarea unui test case pentru editarea profilului

În figura 5 putem observa un test case pentru editarea profilului. Pentru aceasta a fost accesat meniul respectiv și modificate numele profilului, locația indicată și descrierea utiliatorului.

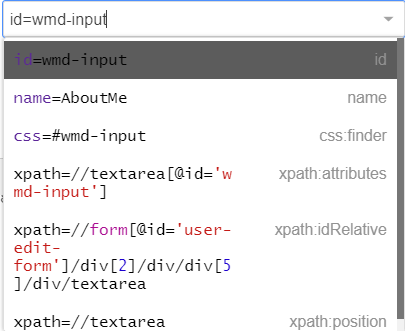


Figura 6 – Meniul drop al comenzii Target

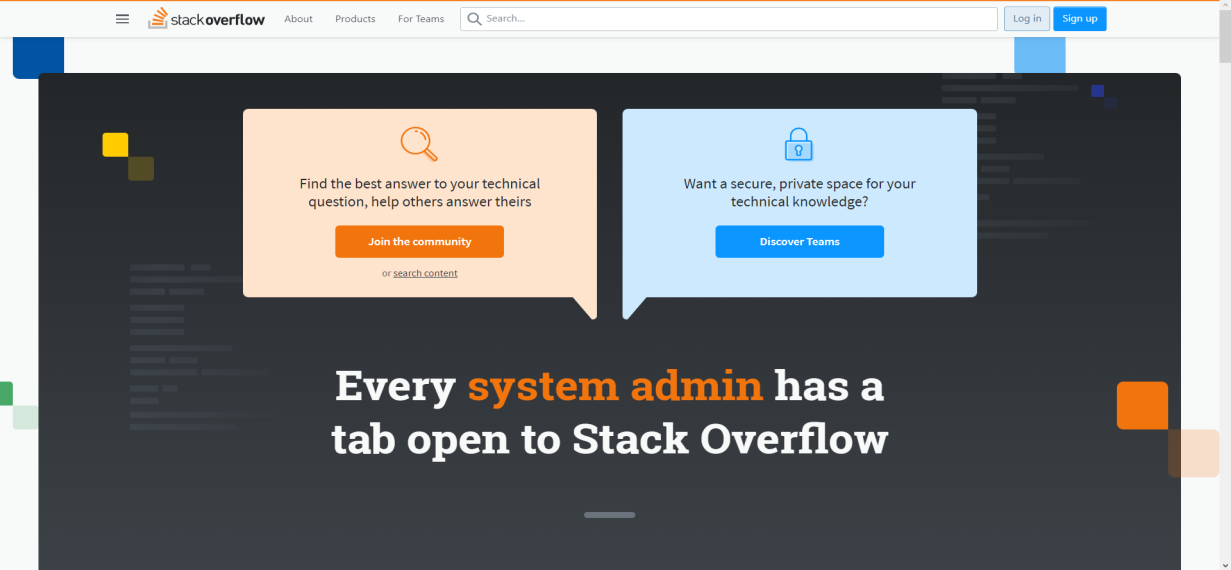
În figura 6 este reprezentat meniul drop al comenzii Target pentru pasul 13 reprezentat în figura 5. După schimbarea setării default cu oricare din cele demonstrate, rezultatul final nu s-a schimbat. Setarea dată reperezintă mai multe modalități de a indica targetul dorit.

Figura 7 – Pagina testată

**Concluzie:**

În concluzie, după realizarea acestui laborator am utilizat cunoștințele acumulate anterior pentru a dezvolta două teste pentru pagini web folosind aplicația Selenium. Testele au fost efectuate cu succes și a fost evaluată performanța aplicației. Am creat un caz de test pentru site-ul web StackOverflow, în care am examinat funcționalitățile de bază ale site-ului. Această experiență acumulată va reprezenta o bază solidă de cunoștințe pentru testare.

**Bibliografie:**

1. Selenium IDE - <https://www.selenium.dev>

2. Pagina testată - <https://stackoverflow.com>

**Anexa:**

// Generated by Selenium IDE

import org.junit.Test;

import org.junit.Before;

import org.junit.After;

import static org.junit.Assert.\*;

import static org.hamcrest.CoreMatchers.is;

import static org.hamcrest.core.IsNot.not;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.remote.RemoteWebDriver;

import org.openqa.selenium.remote.DesiredCapabilities;

import org.openqa.selenium.Dimension;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

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

import org.openqa.selenium.support.ui.WebDriverWait;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.Alert;

import org.openqa.selenium.Keys;

import java.util.\*;

import java.net.MalformedURLException;

import java.net.URL;

public class UntitledTest {

  private WebDriver driver;

  private Map<String, Object> vars;

  JavascriptExecutor js;

  @Before

  public void setUp() {

    driver = new ChromeDriver();

    js = (JavascriptExecutor) driver;

    vars = new HashMap<String, Object>();

  }

  @After

  public void tearDown() {

    driver.quit();

  }

  @Test

  public void untitled() {

    driver.get("https://www.calculator.net/bmr-calculator.html?ctype=metric&cage=25&csex=m&cheightfeet=5&cheightinch=10&cpound=160&cheightmeter=180&ckg=60&cmop=0&coutunit=c&cformula=m&cfatpct=20");

    driver.manage().window().setSize(new Dimension(1152, 626));

    driver.findElement(By.cssSelector(".clearbtn")).click();

    driver.findElement(By.id("cage")).click();

    driver.findElement(By.id("cage")).sendKeys("21");

    driver.findElement(By.cssSelector("td > .cbcontainer:nth-child(1)")).click();

    driver.findElement(By.id("cheightmeter")).click();

    driver.findElement(By.id("cheightmeter")).sendKeys("185");

    driver.findElement(By.id("ckg")).click();

    driver.findElement(By.id("ckg")).sendKeys("70");

    driver.findElement(By.linkText("+ Settings")).click();

    driver.findElement(By.cssSelector("#ccsettingcontent > div > .cbcontainer:nth-child(1)")).click();

    driver.findElement(By.cssSelector("div:nth-child(2) > .cbcontainer")).click();

    driver.findElement(By.cssSelector("tr:nth-child(2) > td:nth-child(1) > input")).click();

    driver.get("https://stackoverflow.com");

    assertThat(driver.findElement(By.linkText("Products")).getText(), is("Product"));

    assertThat(driver.findElement(By.linkText("Products")).getText(), is("Products"));

    vars.put("txtStored", driver.findElement(By.cssSelector(".bg-white .js-billed-annually")).getText());

    System.out.println(vars.get("txtStored").toString());

    System.out.println("$6.00");

  }

}