|  |  |
| --- | --- |
| Test ID: | 1.1 |
| Area of functionality: | Modul za registraciju korisnika |
| Objective | Provjera funkcionalnosti registracije novog korisnika u sustav |
| Test case results: | 6 testnih točki: 6 Pass, 0 Fail |
| Note: | Korisnik nije registriran |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test seq. ID | Action taken | Prediction | Expected results | Pass/fail | Note |
| 1. | U google tražilicu upisati web adresu: https://demo.opencart.com/ | - | Korisnik je uspješno pristupio početnoj stranici | Pass |  |
| 2. | U footeru korisnik odabire opciju My Account | - | Korisnik je uspješno pristupio stranici za Prijavu/registraciju | Pass |  |
| 3. | Korisnik odabire registraciju računa klikom na gumb Continue | Korisnik prethodno nije registriran s mail adresom | Korisnik je uspješno pristupio stranici za unos svojih podataka | Pass |  |
| 4. | Korisnik popunjava sve obvezne podatke i označava checkbox 'I have read and agree to the Privacy Policy.' | Unesena email adresa ne postoji u bazi podataka | Korisnik je ispravno unio podatke. | Pass |  |
| 5. | Korisnik je potvrdio unseen podatke klikom na gumb Continue | Korisnik je prihvatio uvjete korištenja | Korisnik dobiva obavijest o uspješnoj registraciji. | Pass |  |
| 6. | Korisnik klikom na Continue dobiva pregleda računa | - | Korisnik ima mogućnost upravljanja korisnickim računom | Pass |  |

using System;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading;

using NUnit.Framework;

using OpenQA.Selenium;

using OpenQA.Selenium.Firefox;

using OpenQA.Selenium.Support.UI;

namespace SeleniumTests

{

[TestFixture]

public class UntitledTestCase

{

private IWebDriver driver;

private StringBuilder verificationErrors;

private string baseURL;

private bool acceptNextAlert = true;

[SetUp]

public void SetupTest()

{

driver = new FirefoxDriver();

baseURL = "https://www.google.com/";

verificationErrors = new StringBuilder();

}

[TearDown]

public void TeardownTest()

{

try

{

driver.Quit();

}

catch (Exception)

{

// Ignore errors if unable to close the browser

}

Assert.AreEqual("", verificationErrors.ToString());

}

[Test]

public void TheUntitledTestCaseTest()

{

driver.Navigate().GoToUrl("https://demo.opencart.com/index.php?route=common/home");

driver.FindElement(By.XPath("//a[contains(text(),'My Account')]")).Click();

driver.FindElement(By.LinkText("Continue")).Click();

driver.FindElement(By.Id("input-firstname")).Click();

driver.FindElement(By.Id("input-firstname")).Clear();

driver.FindElement(By.Id("input-firstname")).SendKeys("Testo");

driver.FindElement(By.Id("input-lastname")).Click();

driver.FindElement(By.Id("input-lastname")).Clear();

driver.FindElement(By.Id("input-lastname")).SendKeys("Testic");

driver.FindElement(By.Id("input-email")).Click();

driver.FindElement(By.Id("input-email")).Clear();

driver.FindElement(By.Id("input-email")).SendKeys("testo.testic1111@gmail.com");

driver.FindElement(By.Id("input-telephone")).Click();

driver.FindElement(By.Id("input-telephone")).Clear();

driver.FindElement(By.Id("input-telephone")).SendKeys("0999991111");

driver.FindElement(By.Id("input-password")).Click();

driver.FindElement(By.Id("input-password")).Clear();

driver.FindElement(By.Id("input-password")).SendKeys("testotestic111");

driver.FindElement(By.Id("input-confirm")).Click();

driver.FindElement(By.Id("input-confirm")).Clear();

driver.FindElement(By.Id("input-confirm")).SendKeys("testotestic111");

driver.FindElement(By.Name("agree")).Click();

driver.FindElement(By.XPath("//input[@value='Continue']")).Click();

driver.FindElement(By.LinkText("Continue")).Click();

}

private bool IsElementPresent(By by)

{

try

{

driver.FindElement(by);

return true;

}

catch (NoSuchElementException)

{

return false;

}

}

private bool IsAlertPresent()

{

try

{

driver.SwitchTo().Alert();

return true;

}

catch (NoAlertPresentException)

{

return false;

}

}

private string CloseAlertAndGetItsText() {

try {

IAlert alert = driver.SwitchTo().Alert();

string alertText = alert.Text;

if (acceptNextAlert) {

alert.Accept();

} else {

alert.Dismiss();

}

return alertText;

} finally {

acceptNextAlert = true;

}

}

}

}

|  |  |
| --- | --- |
| Test ID: | 1.2 |
| Area of functionality: | Modul za prijavu korisnika |
| Objective | Provjera funkcionalnosti prijave korisnika u sustav |
| Test case results: | 4 testnih točki: 4 Pass, 0 Fail |
| Note: | Korisnik je registriran |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test seq. ID | Action taken | Prediction | Expected results | Pass/fail | Note |
| 1. | U google tražilicu upisati web adresu: https://demo.opencart.com/ | - | Korisnik je uspješno pristupio početnoj stranici | Pass |  |
| 2. | U footeru korisnik odabire opciju My Account | - | Korisnik je uspješno pristupio stranici za Prijavu/registraciju | Pass |  |
| 3. | Korisnik unosi korisničko ime I lozinku za prijavu | Korisnik je prethodno registriran | Korisnik je uspješno pristupio stranici za unos svojih podataka | Pass |  |
| 4. | Korisnik je potvrdio unesene podatke klikom na Login | Uneseno korisnicko ime I lozinka su ispravni | Korisnik ima mogućnost upravljanja s korisnickim računom | Pass |  |

using System;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading;

using NUnit.Framework;

using OpenQA.Selenium;

using OpenQA.Selenium.Firefox;

using OpenQA.Selenium.Support.UI;

namespace SeleniumTests

{

[TestFixture]

public class UntitledTestCase

{

private IWebDriver driver;

private StringBuilder verificationErrors;

private string baseURL;

private bool acceptNextAlert = true;

[SetUp]

public void SetupTest()

{

driver = new FirefoxDriver();

baseURL = "https://www.google.com/";

verificationErrors = new StringBuilder();

}

[TearDown]

public void TeardownTest()

{

try

{

driver.Quit();

}

catch (Exception)

{

// Ignore errors if unable to close the browser

}

Assert.AreEqual("", verificationErrors.ToString());

}

[Test]

public void TheUntitledTestCaseTest()

{

driver.Navigate().GoToUrl("https://demo.opencart.com/index.php?route=common/home");

driver.FindElement(By.XPath("//a[contains(text(),'My Account')]")).Click();

driver.FindElement(By.Id("input-email")).Click();

driver.FindElement(By.Id("input-email")).Clear();

driver.FindElement(By.Id("input-email")).SendKeys("testo.testic1111@gmail.com");

driver.FindElement(By.Id("input-password")).Click();

driver.FindElement(By.Id("input-password")).Clear();

driver.FindElement(By.Id("input-password")).SendKeys("testo12345");

driver.FindElement(By.XPath("//input[@value='Login']")).Click();

}

private bool IsElementPresent(By by)

{

try

{

driver.FindElement(by);

return true;

}

catch (NoSuchElementException)

{

return false;

}

}

private bool IsAlertPresent()

{

try

{

driver.SwitchTo().Alert();

return true;

}

catch (NoAlertPresentException)

{

return false;

}

}

private string CloseAlertAndGetItsText() {

try {

IAlert alert = driver.SwitchTo().Alert();

string alertText = alert.Text;

if (acceptNextAlert) {

alert.Accept();

} else {

alert.Dismiss();

}

return alertText;

} finally {

acceptNextAlert = true;

}

}

}

}

|  |  |
| --- | --- |
| Test ID: | 1.3 |
| Area of functionality: | Modul za promjenu lozinke |
| Objective | Provjera funkcionalnosti promjene lozinke korisničkog računa |
| Test case results: | 4 testnih točki: 4 Pass, 0 Fail |
| Note: | Korisnik je prijavljen |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test seq. ID | Action taken | Prediction | Expected results | Pass/fail | Note |
| 1. | U google tražilicu upisati web adresu: https://demo.opencart.com/ | - | Korisnik je uspješno pristupio početnoj stranici | Pass |  |
| 2. | U footeru korisnik odabire opciju My Account | Korisnik je prethodno prijavljen | Korisnik je uspješno pristupio svojim podacima | Pass |  |
| 3. | Korisnik je odabrao opciju Change your password | - | Korisnik je uspješno pristupio stranici za promjenu lozinke | Pass |  |
| 4. | Korisnik unosi novu lozinku 2 puta | Lozinke se podudaraju | Korisnik dobija obavijest o uspješnoj promjeni lozinke | Pass |  |

using System;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading;

using NUnit.Framework;

using OpenQA.Selenium;

using OpenQA.Selenium.Firefox;

using OpenQA.Selenium.Support.UI;

namespace SeleniumTests

{

[TestFixture]

public class UntitledTestCase

{

private IWebDriver driver;

private StringBuilder verificationErrors;

private string baseURL;

private bool acceptNextAlert = true;

[SetUp]

public void SetupTest()

{

driver = new FirefoxDriver();

baseURL = "https://www.google.com/";

verificationErrors = new StringBuilder();

}

[TearDown]

public void TeardownTest()

{

try

{

driver.Quit();

}

catch (Exception)

{

// Ignore errors if unable to close the browser

}

Assert.AreEqual("", verificationErrors.ToString());

}

[Test]

public void TheUntitledTestCaseTest()

{

driver.Navigate().GoToUrl("https://demo.opencart.com/index.php?route=account/account");

driver.FindElement(By.LinkText("Change your password")).Click();

driver.FindElement(By.Id("input-password")).Click();

driver.FindElement(By.Id("input-password")).Clear();

driver.FindElement(By.Id("input-password")).SendKeys("testo12345");

driver.FindElement(By.Id("input-confirm")).Click();

driver.FindElement(By.Id("input-confirm")).Clear();

driver.FindElement(By.Id("input-confirm")).SendKeys("testo12345");

driver.FindElement(By.XPath("//input[@value='Continue']")).Click();

}

private bool IsElementPresent(By by)

{

try

{

driver.FindElement(by);

return true;

}

catch (NoSuchElementException)

{

return false;

}

}

private bool IsAlertPresent()

{

try

{

driver.SwitchTo().Alert();

return true;

}

catch (NoAlertPresentException)

{

return false;

}

}

private string CloseAlertAndGetItsText() {

try {

IAlert alert = driver.SwitchTo().Alert();

string alertText = alert.Text;

if (acceptNextAlert) {

alert.Accept();

} else {

alert.Dismiss();

}

return alertText;

} finally {

acceptNextAlert = true;

}

}

}

}