

Universitatea din București

Proiect Testarea Sistemelor Software

Profesor Coordonator:

Predut Sorina-Nicoleta

Studenti participanți:

Stana Marius-Vlad

Bora Dragoș Ionuț

Iancu Florentina-Mihaela

Țugui Iustin-Ion

Cuprins

1. Prezentare Generală Selenium

2. Prezentare Generală Puppeteer

3. Motivarea Alegerii Frameworkului

4. Dezvoltarea testelor

5. Utilizare A.I.

6. Bibliografie

I

PREZENTARE GENERALĂ SELENIUM

Selenium Framework

Selenium este un set de instrumente open-source utilizat pentru automatizarea testelor pe aplicații web. Acest framework oferă un mediu flexibil și puternic pentru dezvoltatorii și testatorii de software, permițându-le să automatizeze interacțiunile cu aplicațiile web într-un mod eficient și robust.

Componentele Selenium

Selenium este alcătuit din mai multe componente esențiale:

Selenium WebDriver: Este componenta principală a framework-ului, care permite controlul automat al browser-ului. WebDriver oferă o interfață simplă pentru a interacționa cu diverse browsere web, precum Chrome, Firefox, Safari, etc.

Selenium IDE (Integrated Development Environment): Este o unealtă de dezvoltare a testelor care permite înregistrarea și reproducerea acțiunilor utilizatorului pe o aplicație web. Cu Selenium IDE, testele pot fi create rapid și ușor, fără a fi necesare cunoștințe avansate de programare.

Selenium Grid: Este o componentă care permite rularea testelor pe mai multe mașini și browsere simultan, în paralel. Acest lucru este util pentru testarea în diferite medii și pentru asigurarea compatibilității cross-browser.

Avantaje Selenium

+ **Support Cross-Browser:** Selenium suportă o gamă largă de browsere, inclusiv Chrome, Firefox, Safari, Edge și altele, permițând teste extinse cross-browser.

+ **Support pentru Multiple Limbaje de Programare:** Selenium oferă suport oficial pentru diverse limbaje de programare, inclusiv Java, Python, C#, și altele, fiind potrivit pentru diferiți dezvoltatori.

+ **Comunitate Mare:** Selenium beneficiază de o comunitate mare de utilizatori, ceea ce a dus la o documentație bogată, tutoriale și suport comunitar, facilitând găsirea răspunsurilor la problemele comune.

+ Stabilitate: Selenium există de mult timp și a crescut de-a lungul timpului, devenind o opțiune solidă și stabilă pentru testarea web.

+ Integrare cu Framework-uri de Testare: Selenium se integrează bine cu framework-uri de testare comune precum TestNG, JUnit și NUnit, oferind funcționalități și caracteristici suplimentare.

Dezavantaje Selenium

- Configurare Complexă: Comparativ cu Puppeteer, configurarea Selenium poate fi mai complicată deoarece necesită menținerea driverelor specifice fiecărui browser și asigurarea unor setări potrivite.

- Execuție Mai Lentă: Comparativ cu Puppeteer, Selenium folosește WebDriver, care adaugă un strat suplimentar de comunicare între scriptul de test și browser, rezultând în timpi de execuție puțin mai lenti.

- Probleme de Sincronizare: Din cauza comportamentului asincron al browserelor, teste Selenium pot întâmpina probleme de sincronizare, necesitând așteptări explicite suplimentare sau tehnici de sincronizare.

- Manipularea Ferestrelor Pop-up: Manipularea ferestrelor pop-up sau a alertelor Selenium poate fi dificilă, necesitând cod special de manipulare pentru astfel de situații.

- Suport Inconsistent al Browserelor: În timp ce Selenium suportă o varietate de browsere, cantitatea de suport și funcționalitate poate varia între versiunile browserelor, rezultând în preocupări legate de compatibilitate în anumite cazuri.

Concluzie

Selenium este un instrument esențial în arsenalul de testare a oricărui dezvoltator de software. Prin automatizarea testelor web, Selenium ajută la îmbunătățirea calității software-ului, reducând timpul și costurile asociate testării manuale și asigurând o experiență consistentă pentru utilizatori pe diferite platforme și browsere web.

II

PREZENTARE GENERALĂ PUPPETEER

Puppeteer Framework

Puppeteer este un framework open-source dezvoltat de Google, special conceput pentru automatizarea acțiunilor în browserul Google Chrome. Acest framework oferă un set de instrumente puternice pentru dezvoltatorii și testatorii de software, permițându-le să simuleze interacțiunile utilizatorului și să efectueze teste automate pe aplicații web.

Componentele Puppeteer

Puppeteer API: Este componenta principală a framework-ului, care oferă o interfață de programare pentru controlul automat al browser-ului Chrome. Cu ajutorul API-ului Puppeteer, dezvoltatorii pot naviga pe pagini web, interacționa cu elementele DOM, efectua acțiuni de clic, completare a formularelor, extragere de date și multe altele.

Headless Mode: Puppeteer permite rularea browser-ului Chrome în modul headless, ceea ce înseamnă că nu se va afișa interfața grafică a browser-ului. Acest lucru este util în special pentru testarea automată în medii de integrare continuă sau pentru rularea de scripturi automate pe servere fără interfață grafică.

DevTools Protocol: Puppeteer utilizează protocolul DevTools pentru a comunica cu browser-ul Chrome. Acest lucru oferă acces la o gamă largă de funcționalități avansate, precum monitorizarea rețelei, analiza performanței, debugging-ul aplicațiilor web și multe altele.

Avantaje pentru Puppeteer:

+ **Simplu de Utilizat:** Puppeteer are o interfață API simplă și intuitivă, care îl face ușor de înțeles și de folosit pentru automatizarea sarcinilor în browser.

+ **Produs Google:** Puppeteer este un produs Google care controlează direct browser-ul Chrome, oferind suport puternic pentru noile tehnologii web și capacitățile specifice Chrome.

+ **Viteză Mai Mare de Execuție:** Puppeteer operează la un nivel mai scăzut și are o viteză mai mare de execuție decât Selenium, rezultând în rulări mai rapide ale testelor.

+ **Mod Headless:** Puppeteer oferă modul headless, care permite rularea testelor fără o fereastră de browser vizibilă, făcându-l ideal pentru medii CI/CD.

+ **Funcționalități Încorporate:** Puppeteer include capabilități încorporate pentru colectarea capturilor de ecran și crearea fișierelor PDF, făcându-l ideal pentru testarea vizuală și pregătirea rapoartelor.

Dezavantaje pentru Puppeteer

- **Limitat la Browser-ul Chrome:** Puppeteer este limitat la browser-ul Chrome. Prin urmare, poate să nu fie potrivit pentru testarea cross-browser sau pentru scenarii care necesită examinarea unor comportamente specifice ale browserelor.

- **Curba de Învățare:** În timp ce Puppeteer este în general simplu, documentația API-ului său bogată poate duce la o curbă de învățare ridicată pentru scenarii complexe sau utilizare avansată.

- **Suport Oficial pentru Alte Limbaje de Programare:** Puppeteer oferă suport oficial pentru Node.js, însă pentru alte limbaje de programare, există librării întreținute de comunitate, cu diferite niveluri de suport și funcționalitate.

- **Dependențe Suplimentare:** Puppeteer implică instalarea și gestionarea unor dependențe suplimentare, precum browser-ul Chrome și biblioteca Puppeteer însăși, ceea ce adaugă la complexitatea configurării.

Concluzie

Puppeteer este un instrument puternic pentru automatizarea și testarea aplicațiilor web în browserul Chrome. Cu ajutorul acestui framework, dezvoltatorii pot crea teste automate robuste și eficiente, asigurând calitatea și stabilitatea aplicațiilor lor web într-un mod simplu și elegant.

III

MOTIVAREA ALEGERII FRAMEWORKULUI

Echipa noastră a optat pentru folosirea Selenium Framework deoarece Selenium WebDriver este o alegere excelentă pentru automatizarea testelor într-un proiect web C#. În acest sens, Selenium în cazul nostru are avantajele următoare:

Suport oficial pentru C#: Selenium oferă suport oficial pentru C#, ceea ce facilitează integrarea și dezvoltarea testelor automate în C#.

Integrare ușoară: Întrucât proiectul este în C#, integrarea Selenium WebDriver este simplă și fluentă, permițându-ți să automatizezi interacțiunile cu aplicația web direct din codul C#.

Flexibilitate în scrierea testelor: Selenium oferă o interfață flexibilă pentru interacțiunea cu elementele de pe paginile web, facilitând efectuarea unei varietăți de acțiuni, cum ar fi completarea formularilor sau navigarea între pagini.

Compatibilitate cross-browser: Selenium suportă o gamă largă de browsere, asigurând testarea corectă a aplicației pe diferite platforme și browsere importante.

Suport comunitar activ: Beneficiind de o comunitate mare și activă, Selenium oferă resurse și suport extinse pentru a ne ajuta în dezvoltarea și testarea proiectului nostru web.

IV

DEZVOLTAREA TESTELOR

4.1 Instrumente și Tehnologii Utilizate

Limbaje folosite: Backend: în cadrul proiectului am folosit limbajul C# pentru dezvoltarea aplicației web, iar pentru baza de date a utilizatorilor SQL

Frontend: pe partea de frontend am folosit HTML și CSS

Testare: C#

Framework-uri de Testare: Selenium și Puppeteer

Alte Instrumente și Biblioteci: PuppeteerSharp, Selenium.Chrome.WebDriver, Selenium.WebDriver.

4.2 Abordări de Testare

1. Login și Register pentru Utilizatori:

- **Tip de Testare:** Integrare
- **Descriere:** Verifică autentificarea și înregistrarea utilizatorilor.

2. Search Bar-ul:

- **Tip de Testare:** Unitară
- **Descriere:** Testează funcționarea corectă a căutării.

3. Run Test:

- **Tip de Testare:** Unitară
- **Descriere:** Verifică rularea cu succes a unui test specific.

4. EditUser și DeleteUser:

- **Tip de Testare:** Integrare
- **Descriere:** Verifică editarea și ștergerea profilurilor utilizatorilor.

5. Adăugare Elemente în Watchlist:

- **Tip de Testare:** Integrare
- **Descriere:** Testează adăugarea corectă a elementelor într-o listă de urmărire.

6. Recenzii:

- **Tip de Testare:** Integrare
- **Descriere:** Verifică funcționarea sistemului de recenzii.

7. Achievements

- **Tip de Testare:** Integrare
- **Descriere:** Testează funcționalitatea sistemului de realizări.

8. Quiz Score (Scor în Quiz):

- **Tip de Testare:** Integrare
- **Descriere:** Verifică calculul corect al scorului în quiz.

9. Logout:

- **Tip de Testare:** Integrare
- **Descriere:** Testează deconectarea corectă a utilizatorilor din aplicație.

V

UTILIZAREA A.I.

În cadrul procesului de testare, am încercat și scrierea testelor de către inteligența artificială. În acest sens, am rugat ChatGPT 3.5 să genereze testări pentru funcționalitățile de login, register, respectiv search bar.

Principala observație pe care o avem în lucrul cu ChatGPT este că acesta generează testele corect în procent estimativ de 85%, dar într-un mod prea general. (Acest procent poate scădea în cadrul testelor mai complicate)

5.1 Testare Search Bar

```
public void VerifySearchBar(string movieName)
{
    // Navigate to the movie site
    driver.Navigate().GoToUrl("url_of_your_movie_site");

    // Locate the search bar and input the movie name
    IWebElement searchInput = driver.FindElement(By.Id("search-bar-id"));
    searchInput.SendKeys(movieName);
    searchInput.SendKeys(Keys.Enter);

    // Wait for search results to load
    // You may need to wait for search results to load here

    // Get the title of the first movie in the search results
    IWebElement firstResultTitle = driver.FindElement(By.CssSelector(".search-res

    // Verify if the first movie title corresponds to the provided movie name
    if (firstResultTitle.Text.Equals(movieName, StringComparison.OrdinalIgnoreCas
    {
        Console.WriteLine("Search bar is working. First movie found corresponds t
    }
    else
    {
        Console.WriteLine("Search bar is not working properly. First movie found
    }
}
```

Varianta ChatGPT SearchBar

Astfel, concret vorbind, la testul Search Barului, modificările necesare în cazul nostru au fost:

- Ștergerea parametrului din cadrul funcției.
- Inserarea de mână a numelui filmului pe care îl căutăm.

```
[TestMethod]
public void TestGPT_Register()
{
    // Navigate to the registration page
    driver.Navigate().GoToUrl(baseUrl + "/Utilizatori/Create");

    // Fill in the registration form
    driver.FindElement(By.Name("nume")).SendKeys("John");
    driver.FindElement(By.Name("prenume")).SendKeys("Doe");
    driver.FindElement(By.Id("parola")).SendKeys("Example12//");
    driver.FindElement(By.Name("email")).SendKeys("john.doe@example.com");
    driver.FindElement(By.Id("telefon")).SendKeys("0034567890");

    driver.FindElement(By.Name("data_nastere")).SendKeys("01-01-1990");

    driver.FindElement(By.Name("data_nastere")).SendKeys(Keys.Tab); // Move to the next field
    driver.FindElement(By.Name("data_nastere")).SendKeys("1200PM");

    // Scroll down to the submit button
    IWebElement submitButton = driver.FindElement(By.Name("create_util"));
    ((IJavaScriptExecutor)driver).ExecuteScript("arguments[0].scrollIntoView(true);", submitButton);

    // Submit the form
    // submitButton.Click(); // nu merge versiunea de la GPT
    new Actions(driver).Click(submitButton).Perform();

    // Wait for the main page to load (custom wait condition for URL change)
    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(5000));
    wait.Until(driver => driver.Url.StartsWith(baseUrl + "/"));

    // Assert that the URL matches the main page URL
    Assert.AreEqual(baseUrl + "/", driver.Url);
}
```

Variantă **funcțională**

5.2 Testare Login

Ca și în cazul Search Barului, codul inițial generat a fost satisfăcător, dar prea general.

```
[TestMethod]
public void TestLogin()
{
    // Navigate to the login page
    driver.Navigate().GoToUrl(baseUrl + "/Login");

    // Fill in the login form
    driver.FindElement(By.Id("username")).SendKeys("your_username");
    driver.FindElement(By.Id("password")).SendKeys("your_password");

    // Submit the login form
    driver.FindElement(By.Id("loginButton")).Click();

    // Wait for the main page to load (custom wait condition for URL change)
    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
    wait.Until(driver => driver.Url.StartsWith(baseUrl + "/"));

    // Assert that the URL matches the main page URL
    Assert.AreEqual(baseUrl + "/", driver.Url);
}
```

Schimbările *minimale* necesare au fost:

- Denumirea variabilelor pentru cazul nostru

```
[TestMethod]
| 0 references
public void TestGPT_Login()
{
    // Navigate to the login page
    driver.Navigate().GoToUrl(baseUrl + "/Utilizatori/SignIn");

    // Fill in the login form
    driver.FindElement(By.Id("email")).SendKeys("iancumihaela@gmail.com");
    driver.FindElement(By.Id("parola")).SendKeys("123456");

    // Submit the login form
    driver.FindElement(By.Id("SignIn")).Click();

    // Wait for the main page to load (custom wait condition for URL change)
    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
    wait.Until(driver => driver.Url.StartsWith(baseUrl + "/"));

    // Assert that the URL matches the main page URL
    Assert.AreEqual(baseUrl + "/", driver.Url);
}
```

Variantă **funcțională**

5.3 Testare Register

Spre deosebire de cele două teste menționate anterior, codul generat pentru funcționalitatea de **Register** nu a fost atât de satisfăcător.

```
[TestMethod]
public void TestRegistrationAndReturnToMainPage()
{
    // Navigate to the registration page
    driver.Navigate().GoToUrl(baseUrl + "/Utilizatori/Create");

    // Fill in the registration form
    driver.FindElement(By.Id("lastname")).SendKeys("Doe");
    driver.FindElement(By.Id("firstname")).SendKeys("John");
    driver.FindElement(By.Id("password")).SendKeys("examplepassword");
    driver.FindElement(By.Id("email")).SendKeys("john.doe@example.com");
    driver.FindElement(By.Id("phone")).SendKeys("1234567890");
    driver.FindElement(By.Id("birthday")).SendKeys("1990-01-01"); // Assuming the

    // Submit the form
    driver.FindElement(By.Id("registerButton")).Click();

    // Wait for the main page to load
    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
    wait.Until(ExpectedConditions.UrlToBe(baseUrl + "/"));

    // Assert that the URL matches the main page URL
    Assert.AreEqual(baseUrl + "/", driver.Url);
}
```

Variantă **ChatGPT** Register

În acest caz, până să ajungem la un cod satisfăcător la nivel general, a trebuit să purtăm mai multe dialoguri, în care am cerut:

- Rezolvarea unei erori în legătura cu data
- Scrollarea până la elementul dorit, lucru care nu se întâmpla
- Rezolvarea unei erori în urma click-ului pe buton, aici am suplimentat cu implementarea noastră
- Regenerarea bucății de cod aferente așteptării ca pagina principală să se încarce

Rezultatul final, potrivit în cazul nostru este:

```
[TestMethod]
0 references
public void TestGPT_Register()
{
    // Navigate to the registration page
    driver.Navigate().GoToUrl(baseUrl + "/Utilizatori/Create");

    // Fill in the registration form
    driver.FindElement(By.Name("nume")).SendKeys("John");
    driver.FindElement(By.Name("prenume")).SendKeys("Doe");
    driver.FindElement(By.Id("parola")).SendKeys("Example12//");
    driver.FindElement(By.Name("email")).SendKeys("john.doe@example.com");
    driver.FindElement(By.Id("telefon")).SendKeys("0034567890");

    driver.FindElement(By.Name("data_nastere")).SendKeys("01-01-1990");

    driver.FindElement(By.Name("data_nastere")).SendKeys(Keys.Tab); // Move to the next field
    driver.FindElement(By.Name("data_nastere")).SendKeys("1200PM");

    // Scroll down to the submit button
    IWebElement submitButton = driver.FindElement(By.Name("create_util"));
    ((IJavaScriptExecutor)driver).ExecuteScript("arguments[0].scrollIntoView(true);", submitButton);

    // Submit the form
    // submitButton.Click(); // nu merge versiunea de la GPT
    new Actions(driver).Click(submitButton).Perform();

    // Wait for the main page to load (custom wait condition for URL change)
    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(5000));
    wait.Until(driver => driver.Url.StartsWith(baseUrl + "/"));

    // Assert that the URL matches the main page URL
    Assert.AreEqual(baseUrl + "/", driver.Url);
}
```

Variantă **funcțională**

VI

BIBLIOGRAFIE

- <https://oxylabs.io/blog/puppeteer-vs-selenium>
- <https://brightdata.com/blog/proxy-101/puppeteer-vs-selenium>
- <https://medium.com/@browserscan/which-to-choose-for-web-scraping-selenium-vs-puppeteer-vs-playwright-d007f5ec938e>
- <https://testsigma.com/blog/puppeteer-vs-selenium/>
- <https://www.browserstack.com/guide/puppeteer-vs-selenium>
- <https://www.browserstack.com/guide/selenium-framework>