AZƏRBAYCAN RESPUBLİKASI ELM VƏ TƏHSİL NAZİRLİYİ

AZƏRBAYCAN TEXNİKİ UNİVERSİTETİ

"Kibertəhlükəsizlik" kafedrası

Əlyazması hüququnda

Fərzəlizadə Günel Vüqar qızı

İxtisas: 060632 – "İnformasiya texnologiyaları və sistemləri mühəndisliyi"

İxtisaslaşma: "İnformasiya mühafizəsi və təhlükəsizliyi"

Mövzu: Selenium vasitəsilə veb təhlükəsizlik testlərinin avtomatlaşdırılması

MAGİSTRLİK DİSSERTASİYASI

Elmi rəhbər **r.ü.f.d. dos., R.H.Sadıqova**

"Kibertəhlükəsizlik" kafedrasının müdiri **t.e.d. dos., Y.N.İmamverdiyev**

Bakı-2023

MÜNDƏRİCAT

İXTİSARLARIN SİYAHISI

QA Quality assurance – Keyfiyyət təminatı

CSRF Cross-Site Request Forgery – Saytlararası Tələb Saxtakarlığı

XSS Cross-Site Scripting – Saytlararası Skriptləmə

IDOR Insecure direct object references – Təhlükəsiz Birbaşa Obyekt İstinadları

SQL Structured Query Language – Strukturlaşdırılmış sorğu dili

OWASP Open Worldwide Application Security Project – Ümumdünya Tətbiq Təhlükəsizliyi Layihəsi

PCI DSS The Payment Card Industry Data Security – Standard Ödəniş Kartı Sənayesi Məlumat Təhlükəsizliyi Standartı

GDPR General Data Protection Regulation – Ümumi Məlumatların Qorunması Qaydaları

CPU central processing unit – Mərkəzi prosessor

PII Personal Identifiable Information – Şəxsi müəyyən edilə bilən məlumatlar

W3C The World Wide Web Consortium – Ümumdünya Şəbəkə Konsorsiumu

WCAG Web Content Accessibility Guidelines – Məzmununa Əlçatanlıq Təlimatları

CI/CD continuous integration (CI) and continuous delivery – davamlı inteqrasiya və davamlı çatdırılma

API Application Programming Interface – Tətbiq Proqramlaşdırma İnterfeysi

HTML Hypertext Markup Language – hipermətn işarələ dili

URL Uniform Resource Locator Uniform – Mənbə Lokatoru

UI User interface – İstifadəçi interfeysi

UX User experience – İstifadəçi təcrübəsi

GİRİŞ

Mövzunun aktuallığı. Bu dissertasiya proqram təminatının keyfiyyətinin təmin edilməsi kontekstində Selenium veb testinin aktuallığını araşdırır. Veb tətbiqetmələrinin sürətli böyüməsi ilə effektiv və etibarlı test metodologiyalarına ehtiyac hər şeydən üstündür. Veb avtomatlaşdırılması üçün geniş şəkildə istifadə edilən açıq mənbəli vasitə olan Selenium, bir çox brauzer və platformalarda veb proqramların sınaqdan keçirilməsi üçün hərtərəfli çərçivə təklif edir. Bu dissertasiya Selenium veb testinin müxtəlif aspektlərini və proqram təminatının keyfiyyət təminatı prosesində əhəmiyyətini araşdırır. Tədqiqat yüksək keyfiyyətli veb tətbiqlərinə nail olmaqda Seleniumun effektivliyini və aktuallığını nümayiş etdirmək üçün empirik tədqiqatlar, nümunə araşdırmaları və sənayenin ən yaxşı təcrübələrini birləşdirir.

Tədqiqatın məqsədi. Tədqiqatın məqsədi proqram təminatının keyfiyyətinin təminatı kontekstində Selenium veb testinin effektivliyini və aktuallığını araşdırmaqdır. Tədqiqatın məqsədi Selenium veb testinin veb əsaslı tətbiqlərin funksionallığının, performansının və istifadəçi təcrübəsinin yaxşılaşdırılmasına təsirini qiymətləndirməkdir. Bundan əlavə, tədqiqatın məqsədi Selenium veb testini onun güclü tərəflərini, məhdudiyyətlərini və potensial təkmilləşdirmə sahələrini müəyyən etmək üçün digər test yanaşmaları ilə müqayisə etməkdir.

Tədqiqatın obyekti və metodologiyası. Tədqiqat proqram təminatının keyfiyyət təminatı proseslərinin təkmilləşdirilməsində Selenium veb testinin effektivliyinin qiymətləndirilməsinə yönəlib. Tədqiqat metodologiyası proqram təminatının keyfiyyətinin təminatı təcrübələrində Selenium veb testinin effektivliyi və aktuallığının hərtərəfli başa düşülməsini təmin etmək üçün empirik təhlil, eksperiment və işin qiymətləndirilməsinin birləşməsini əhatə edir.

Tədqiqatın elmi yeniliyi və praktik əhəmiyyəti. Tədqiqat İnnovasiyası. Selenium veb testi ilə bağlı tədqiqat test proseslərini təkmilləşdirmək və veb-əsaslı tətbiqlərin ümumi keyfiyyətini yaxşılaşdırmaq üçün innovativ yolları araşdıraraq proqram təminatının keyfiyyətinin təminatı sahəsinə töhfə verir. Seleniumun açıq mənbəli avtomatlaşdırma vasitəsi kimi istifadəsi bir sıra innovativ aspektləri təklif edir,

o cümlədən: Çarpaz Brauzer və Platformalar Arası Uyğunluq, Davamlılıq, Davamlı İnteqrasiya/Çatdırılma, Praktik Tətbiq, Təkmilləşdirilmiş Test əhatə dairəsi, Təkmilləşdirilmiş Proqram Keyfiyyəti, Avtomatlaşdırmanın Səmərəliliyi, Xərc və vaxta qənaət, Real-dünya Case Studies, və s. Ümumilikdə, Selenium veb testi ilə bağlı tədqiqatın praktiki tətbiqi proqram təminatının daha səmərəli və effektiv sınaq proseslərinə, proqram təminatının keyfiyyətinin yaxşılaşmasına və proqram təminatı inkişaf sənayesindəki təşkilatlar üçün məhsuldarlığın artmasına səbəb ola bilər.

İşin aprobasiyası. Selenium avtomatlaşdırma testi haqqında bəhs olunub. Bu test üsulları veb test üsulları adlanır. Keyfiyyətin təminatı istər manual, istər də avtomatik test olunur. Selenium isə məhv avtomatik test üsullarından biridir.

Veb tətbiqetməni sınaqdan keçirməzdən əvvəl nəzərə alınmalı olan bəzi faktlar var. Məsələn, tətbiqin hansı hissəsinin avtomatlaşdırılacağı, istifadəçi gözləntiləri, layihə üçün verilən vaxt və prioritetlər layihə meneceri tərəfindən müəyyən edilirdi.

Tədqiqat işinin məqsədi və vəzifələri. Tədqiqat işinin əsas məqsədi Selenium veb testing ilə axtarış, login proseslərinin avtomatlaşdırılması, testinq ilə təhlükəsizliyin avtomatlaşdırılmasıdır. Bu məqsədə çatmaq üçün isə aşağıdakı əsas vəzifələrin icra olunması məqsədə müvafiqdir:

1. QA avtomatlaşdırılmasının öyrənilməsi.
2. Veb brauserlərdə sistemlərdə testetmənin həyata keçirilməsi zamanı istifadə edilən üsulların tədqiqi və müqayisəli analiz edilməsi.
3. Selenium ilə brauzerdə testetmənin reallaşdırılması, nümunə proqram təminatının hazırlanması.

Tədqiqat işinin elmi yeniliyi. Tədqiqat işinin elmi yeniliyini aşağıdakılar təşkil edir:

1. Mövcud üsullarda bəzi çatışmazlıqların - müasir sistemlərdə olan tələblərin ödənilmədiyi halların, aradan qaldırılması istiqamətində həll yolları.
2. Mövcud testetmə üsullarının müsbət və mənfi tərəflərini müəyyən etmək və uyğun tətbiq yerlərinin müəyyən edilməsi.

Tədqiqat işinin strukturu və quruluşu. Dissertasiya işi giriş, 3 fəsil, nəticə və 28 ədəbiyyat mənbəyindən ibarət olmaqla 61 səhifədə təşkil olunmuşdur. İşdə 19 şəkil yer almışdır.

Birinci fəsildə veb tətbiqi testləri, tərifi və cəlb olunan problemlər haqqında məlumat təqdim edir.

İkinci fəsildə proqram təminatının sınaqdan keçirilməsinin əsaslarını və avtomatlaşdırılmış sınaq konsepsiyalarını əhatə edən nəzəri çərçivəni yaradır. Seleniumun üstünlükləri, mənfi cəhətləri və tətbiqi davranışı və sistem tələbləri ilə birlikdə hərtərəfli müzakirə olunur. Seleniumun Python proqramlaşdırma dili ilə inteqrasiyası araşdırılır.

Üçüncü fəsildə veb proqramların sınaqdan keçirilməsində Seleniumun tətbiqini nümayiş etdirmək üçün praktiki nümunələr təqdim olunur. Nümunələrə Seleniumdan istifadə edərək mətn oxuma testinin aparılması və giriş testi ssenarilərinin aparılması daxildir. Bu dissertasiyada təqdim olunan tapıntılar və nümunələr veb proqram sistemlərinin keyfiyyətini və etibarlılığını artıraraq, veb tətbiqetmələrin sınaqdan keçirir.

Minnətdarlıq. Tədqiqat işində göstərdiyi dəstəyə görə “Kibertəhlükəsizlik” kafedrasına və elmi rəhbərim r.ü.f.d. dos., R.H.Sadıqovaya təşəkkür edirəm.

I FƏSİL. VEB TƏTBİQ TESTERİ İLƏ AVTOMATLAŞDRIMA TESTLƏRİ

1.1. Veb tətbiq sınaqlarına ümumi baxış

Veb tətbiqi testi veb proqramın funksionallığını, istifadəsini, təhlükəsizliyini və performansının istənilən tələblərə cavab verməsini təmin etmək üçün qiymətləndirmə prosesidir. Bu, hər hansı qüsurları, səhvləri, zəiflikləri və ya təkmilləşdirilməli sahələri müəyyən etmək üçün veb tətbiqinin müxtəlif komponentlərini və xüsusiyyətlərini sistematik şəkildə araşdırılmasını və təsdiqlənməsini əhatə edir. Veb proqram testinin məqsədi tətbiqin nəzərdə tutulduğu kimi işləməsini, qənaətbəxş istifadəçi təcrübəsini təmin etməsini və tələb olunan keyfiyyət standartlarına cavab verməsini təmin etməkdir. Veb tətbiq testi bir neçə səbəbə görə böyük əhəmiyyət kəsb edir:

Funksionallığın təmin edilməsi: Test veb proqramın nəzərdə tutulduğu kimi işləməsini və müəyyən edilmiş tələblərə cavab verməsini təmin edir.

İlk müsbət cəhəti istifadəçi təcrübəsinin təkmilləşdirilməsidir. İstifadəçi təcrübəsinin təkmilləşdirilməsi: Test istifadəçi təcrübəsinə mane ola biləcək istifadəlik problemlərini və istifadəçi interfeysi problemlərini müəyyən etməyə kömək edir.

Bir sonrakı müsbət cəhət təhlükəsizliyin təmin edilməsi: Veb proqramları tez-tez zəifliklərdən istifadə etmək istəyən zərərli hakerlər tərəfindən hədəfə alınır.

Performansın Optimizasiyası: Performans testi müxtəlif yükləmə şərtləri altında veb tətbiqinin həssaslığını, miqyaslanmasını və sabitliyini qiymətləndirir.

Uyğunluq və Cross-Brauzer Dəstəyi: Veb tətbiqləri bir çox brauzer, əməliyyat sistemi və cihazda işləməlidir.

Reqressiyanın qarşısının alınması: Veb tətbiqləri inkişaf etdikcə və yeni funksiyalar əlavə olunduqca, reqressiya testinin aparılması çox vacibdir.

Standartlara uyğunluq: Sənaye və hədəf auditoriyadan asılı olaraq, veb proqramlar xüsusi standartlara, qaydalara və ya əlçatanlıq qaydalarına uyğun gəlməlidir.

Reputasiya və Müştəri Məmnuniyyətinin Qorunması: Yüksək keyfiyyətli veb tətbiqi istifadəçi məmnuniyyətini artırır, inamı artırır və təşkilatın reputasiyasını artırır.

Bundan əlavə müsbət cəhətlərdə ən əsası xərcə və vaxta qənaətdir. Xərc və vaxta qənaət: Sınaq mərhələsində problemlərin müəyyən edilməsi və həll edilməsi, tətbiqin tətbiqindən sonra onları həll etməkdən daha sərfəli olur.

Xülasə olaraq, veb proqramların testi veb proqramların funksionallığını, istifadəsini, təhlükəsizliyini, performansını və uyğunluğunu təmin etəmin edir, etibarı artırır və hüquqi və ya tənzimləyici problemlərdən qaçınır.

Veb tətbiqi testinin əsas aspektlərinə ümumi baxış: Funksional sınaq, Usability Test, Təhlükəsizlik Testi, Performans Testi, Uyğunluq Testi, Reqressiya Testi, Yük testi, Stress Testi, Əlçatanlıq Testi, Lokallaşdırma Testi.

1. Funksional sınaq: Bu növ sınaq veb proqramın nəzərdə tutulduğu kimi işləməsini yoxlamağa yönəlib. Funksional test, tətbiqin funksional davranışını yoxlamağa yönəlmiş veb tətbiq testinin bir növüdür. O, veb proqramın nəzərdə tutulduğu kimi işləməsini və müəyyən edilmiş tələblərə cavab verməsini təmin edir. Funksional sınaq adətən aşağıdakı fəaliyyətləri əhatə edir:

Test işi dizaynı: Test nümunələri veb tətbiqinin müxtəlif funksiyalarını və ssenarilərini əhatə etmək üçün hazırlanmışdır. Testin icrası: Hazırlanmış test nümunələri girişləri təmin etməklə və gözlənilən nəticələrə qarşı faktiki çıxışları yoxlamaqla yerinə yetirilir. Funksional əhatə dairəsi: Funksional sınaq tətbiqin xüsusiyyətləri və funksiyalarının maksimum əhatəsinə nail olmaq məqsədi daşıyır.

Sərhəd Dəyəri (Boundary Value Analysis testing) Təhlili: Bu texnika tətbiqin giriş dəyərlərinin sərhədlərində davranışının sınaqdan keçirilməsini nəzərdə tutur. Səhvlərin idarə edilməsi: Funksional sınaq tətbiqin səhvləri və istisnaları zərif şəkildə idarə etmək qabiliyyətini yoxlamağı əhatə edir. İnteqrasiya Testi: İnteqrasiya testi veb tətbiqinin müxtəlif komponentlərinin/modullarının qarşılıqlı əlaqəsini və uyğunluğunu yoxlamaq üçün həyata keçirilir. Reqressiya Testi: Reqressiya testi mövcud funksiyalara mənfi təsir göstərmədiyinə əmin olmaq üçün tətbiqdə dəyişikliklər və ya təkmilləşdirmələr edildikdən sonra aparılır. Uyğunluq Testi: Uyğunluq testi veb

tətbiqinin müxtəlif brauzerlər, əməliyyat sistemləri və cihazlarda düzgün işləməsini təmin etmək üçün həyata keçirilir.

Tərtibatçılar və QA komandaları hərtərəfli funksional testlər həyata keçirməklə, yüksək keyfiyyətli istifadəçi təcrübəsini təmin etməklə, veb tətbiqetmədə istehsala yerləşdirilməzdən əvvəl hər hansı funksional qüsurları və ya problemləri müəyyən edib aradan qaldıra bilər.

2. Usability Test: İstifadə qabiliyyəti testi veb tətbiqinin istifadəçiyə uyğunluğunu qiymətləndirir.

İstifadə qabiliyyəti testi: İstifadəyə yararlılıq testi, tətbiqin istifadəçi dostu və istifadəyə yararlılıq aspektlərinin qiymətləndirilməsinə yönəlmiş bir veb tətbiqi test növüdür. Məqsəd veb tətbiqinin intuitiv olmasını, istifadəsi asan olmasını və müsbət istifadəçi təcrübəsini təmin etməkdir. İstifadəyə yararlılıq testinin əsas aspektləri bunlardır:

Testin Planlaşdırılması: İstifadəyə yararlılıq testi istifadəçilərin test zamanı yerinə yetirəcəkləri məqsədləri, hədəf auditoriyanı, ssenariləri və tapşırıqları müəyyən etməklə planlaşdırılır. Test mühiti: İstifadəyə yararlılıq testi adətən lazımi avadanlıq və alətlərlə idarə olunan mühitdə aparılır. İstifadəçilərin işə qəbulu: Veb tətbiqinin hədəf auditoriyasını təmsil edən iştirakçılar istifadəyə yararlılıq testi üçün seçilir. Testin icrası: İstifadə qabiliyyəti testi zamanı iştirakçılara qarşılıqlı əlaqə, müşahidələr və rəylər qeydə alınarkən veb proqramda yerinə yetirmək üçün xüsusi tapşırıqlar verilir. Müşahidələr və Metriklər: İstifadəyə yararlılıq testinin aparıcısı iştirakçıların qarşılıqlı əlaqəsini yaxından izləyir və onların qarşılaşdıqları hər hansı problem, çətinlik və ya çaşqınlıq sahələrini qeyd edir. Təhlil və Hesabat: İstifadəyə yararlılıq testindən toplanmış məlumatlar təkrarlanan nümunələri, istifadəyə yararlılıq problemlərini və təkmilləşdirmə sahələrini müəyyən etmək üçün təhlil edilir.

İstifadə qabiliyyətinin yoxlanılması istifadəçilərin veb tətbiqi ilə necə qarşılıqlı əlaqədə olduğuna dair dəyərli anlayışlar təqdim edir və inkişaf prosesinin əvvəlində potensial istifadə problemlərini müəyyən etməyə kömək edir.

3.Təhlükəsizlik Testi: Təhlükəsizlik testi təcavüzkarlar tərəfindən istifadə edilə bilən veb proqramdakı zəiflikləri müəyyən etmək məqsədi daşıyır.

Təhlükəsizlik testi tətbiqdəki zəiflikləri, boşluqları və potensial təhlükəsizlik risklərini müəyyən etməyə yönəlmiş veb tətbiqi testinin mühüm aspektidir. Təhlükəsizlik testinin əsas aspektləri bunlardır:

Zəifliyin Qiymətləndirilməsi: Bu, məlum təhlükəsizlik zəifliklərini və boşluqlarını müəyyən etmək üçün veb proqramın skan edilməsini nəzərdə tutur. Nüfuz Testi (Penetration Testing): Tez-tez "pen test" olaraq adlandırılan nüfuz testi təhlükəsizlik zəifliklərini aşkar etmək üçün veb tətbiqinə real dünya hücumlarını simulyasiya edir. Doğrulama və Avtorizasiya Testi: Bu, tətbiqin autentifikasiya və avtorizasiya mexanizmlərinin effektivliyinin qiymətləndirilməsini əhatə edir. Sessiyanın İdarə Edilməsi Testi: Sessiyanın idarə edilməsi testi veb tətbiqinin istifadəçi seanslarını düzgün idarə etməsini və sessiyanın qaçırılması və ya sessiyanın fiksasiyası kimi sessiya ilə əlaqəli zəifliklərdən qorumasını təmin etmək məqsədi daşıyır. Daxiletmə Qiymətləndirmə Testi: Daxiletmənin yoxlanılması testi veb tətbiqinin müxtəlif növ istifadəçi daxiletmələrini təhlükəsiz idarə etmək qabiliyyətini yoxlayır. Təhlükəsizlik Konfiqurasiya Testi: Bu, veb server konfiqurasiyaları, verilənlər bazası konfiqurasiyaları, şifrələmə protokolları və şəbəkə təhlükəsizlik parametrləri daxil olmaqla, proqramın təhlükəsizlik konfiqurasiya parametrlərinin nəzərdən keçirilməsini və sınaqdan keçirilməsini əhatə edir. Səhvlərin idarə edilməsi və qeydiyyatın yoxlanılması: Bu, tətbiqin səhvləri təhlükəsiz idarə etdiyini və problemlərin aradan qaldırılması və analiz üçün müvafiq məlumatları qeyd etdiyini yoxlamağı əhatə edir. Təhlükəsizliyə Uyğunluq Testi: Təhlükəsizliyə uyğunluq testi veb tətbiqinin sənaye standartlarına, qaydalara və təhlükəsizliklə bağlı ən yaxşı təcrübələrə uyğunluğunu qiymətləndirir, məsələn OWASP (Açıq Veb Tətbiqi Təhlükəsizliyi Layihəsi) Top 10, Ödəniş Kartı Sənayesi Məlumat Təhlükəsizliyi Standartı (PCI DSS), və ya Ümumi Məlumatların Qorunması Qaydası (GDPR). Tətbiqin tələb olunan təhlükəsizlik standartlarına cavab verməsini təmin edir.

Hərtərəfli təhlükəsizlik testi keçirməklə təşkilatlar təhlükəsizlik zəifliklərini və risklərini müəyyən edib aradan qaldıra, istifadəçi məlumatlarını qoruya və veb proqramın bütövlüyünü və məxfiliyini qoruya bilər.

4.Performans Testi: Performans testi müxtəlif yükləmə şərtləri altında veb tətbiqinin həssaslığını, miqyaslanmasını və sabitliyini qiymətləndirir.

Performans Testi: Performans testi müxtəlif yükləmə şəraitində tətbiqin performansını, miqyaslanmasını və cavab vermə qabiliyyətini qiymətləndirməyə yönəlmiş bir veb tətbiqi test növüdür. Performans testinin əsas aspektləri bunlardır:

Yük sınağı: Yük testi, performansını qiymətləndirmək üçün veb tətbiqini real istifadəçi yüklərinə məruz qoymağı əhatə edir. Stress Testi: Stress testi, normal tutumundan kənarda, həddindən artıq yük şəraitində veb tətbiqinin işini qiymətləndirir. Ölçüləbilən Testi (Scalability test): Ölçüləbilən testi serverlər, verilənlər bazası və ya şəbəkə bant genişliyi (network bandwidth) kimi daha çox resurs əlavə etməklə veb tətbiqinin artan istifadəçi yüklərini idarə etmək qabiliyyətini qiymətləndirir. Performans Metrikləri: Performans testi cavab müddəti, ötürmə qabiliyyəti, gecikmə, CPU və yaddaş istifadəsi, şəbəkə sürəti və verilənlər bazası sorğusunun performansı kimi müxtəlif performans göstəricilərinin ölçülməsini əhatə edir. Performansın ilkin göstəriciləri: Performansın ilkin göstəriciləri məqbul performans hədlərini və meyarları müəyyən etmək üçün ilkin performans testlərinin aparılması ilə müəyyən edilir. Real-Dünya Ssenariləri: Performans testi faktiki istifadə nümunələrini təqlid etmək üçün real dünya ssenarilərini və istifadəçi davranışlarını simulyasiya etməyi əhatə edir. Performans tənzimləməsi: Performans testi performans bottlenecks və optimallaşdırma tələb edən sahələri müəyyən etməyə kömək edir. Dözümlülük Testi: Dözümlülük testi, veb tətbiqini uzun müddət davamlı yükə məruz qoymağı əhatə edir.

Hərtərəfli performans testi keçirməklə təşkilatlar veb tətbiqinin gözlənilən istifadəçi yüklərini idarə edə bilməsini, sürətli cavab vaxtlarını təmin edə və yüksək performans və istifadəçi məmnuniyyətini təmin edə bilər.

5.Uyğunluq Testi: Uyğunluq testi veb tətbiqinin müxtəlif brauzerlər, əməliyyat sistemləri, cihazlar və ekran ölçüləri arasında düzgün işləməsini təmin edir. QA avtomatlaşdırmasında uyğunluq testini həyata keçirərkən nəzərə alınmalı əsas aspektlər bunlardır:

Tətbiq olunan Qaydaları Müəyyən olunma: Proqram sisteminizə tətbiq olunan müvafiq qaydaları, standartları və təlimatları müəyyən olunmalıdır. Uyğunluq

Tələbləri: Tətbiq olunan qaydalarla müəyyən edilmiş xüsusi uyğunluq tələbləri dərindən dərk edilməlidir. Uyğunluq Testinin Planlaşdırılması: Uyğunluq tələblərini və sınaq məqsədlərini əks etdirən hərtərəfli sınaq planı yaradılmalıdır. Test Məlumat Təhlükəsizliyi: Proqram təminatının həssas məlumatları təhlükəsiz şəkildə idarə etdiyini və məlumatların qorunması, şifrələmə və girişə nəzarət tələblərinə cavab verdiyi yoxlanmalıdır. Girişə Nəzarət və İstifadəçi İcazələri: Müvafiq istifadəçi rollarının və icazələrin həyata keçirilməsini təmin edərək proqram təminatının girişə nəzarət mexanizmlərini yoxlanmalıdır. Məxfilik və Razılığın İdarə Edilməsi: Proqram təminatının məlumatların toplanması, istifadəsi və paylaşılması üçün istifadəçi razılığının alınması kimi məxfilik qaydalarına uyğunluğunu qiymətləndirmək lazımdır. Audit Trails və Logging: Proqram təminatının istifadəçi fəaliyyətlərinin və sistem dəyişikliklərinin audit izini saxladığını yoxlamaq lazımdır. Təhlükəsizlik Testi: Nüfuz testi, zəifliyin skan edilməsi və təhlükəsizlik kodunun nəzərdən keçirilməsi kimi zəiflikləri müəyyən etmək üçün təhlükəsizlik testi lazımdır. Əlçatanlıq Testi: WCAG (Veb Məzmununa Əlçatımlılıq Təlimatları) kimi əlçatanlıq standartlarına uyğunluğu təmin etmək üçün proqram təminatının əlçatanlıq xüsusiyyətlərini yoxlamaq lazımdır. Uyğunluq Hesabatı: Uyğunluq testinin nəticələrini sənədləşdirən və uyğunluğun sübutunu təqdim edən hərtərəfli hesabatlar yaradılır. Davamlı Uyğunluğun Monitorinqi: Müvafiq qaydalara davamlı olaraq nəzarət etmək və davamlı uyğunluğu təmin etmək üçün prosesləri həyata keçirilir.

Bu əsas aspektləri həll etməklə, QA avtomatlaşdırmasında uyğunluq testi təşkilatlara proqram sistemlərinin lazımi tənzimləyici və sənayeyə aid tələblərə cavab verməsini təmin etməyə kömək edir, onların standartlara və təlimatlara uyğunluğuna əminlik yaradır.

Hər bir performans testi ilə təşkilatlar veb tətbiqinin gözlənilən yüklərini idarə edə bilməsini, sürətli cavablarını təmin etmək və yüksək performans və səmərəliliyi təmin etmək.

6.Reqressiya Testi: Reqressiya testi tətbiqdə edilən dəyişikliklərin və ya düzəlişlərin yeni səhvlər və ya problemlər təqdim etməməsini təmin etmək üçün əvvəllər sınaqdan keçirilmiş funksiyaların yenidən sınaqdan keçirilməsini əhatə edir.

Reqressiya testi, dəyişikliklər və ya təkmilləşdirmələr edildikdən sonra tətbiqin mövcud funksiyalarının toxunulmaz və təsirsiz qaldığını yoxlamağa yönəlmiş veb tətbiqi testinin bir növüdür. Reqressiya testinin əsas aspektləri bunlardır:

Test İşinin Seçilməsi: Reqressiya testi mövcud test dəstindən veb tətbiqinin kritik funksiyalarını əhatə edən test işlərinin bir hissəsinin seçilməsini əhatə edir. Test işlərinin prioritetləşdirilməsi: Test işlərinin prioritetləşdirilməsi onların əhəmiyyətinə və tətbiqin funksionallığına təsirinə əsasən verilir. Test Suite (dəstinə) qulluq: Test dəsti dəyişdirilmiş və ya əlavə edilmiş funksiyalar üçün yeni test nümunələri daxil etmək üçün yenilənir və saxlanılır. Versiyaya Nəzarət və Sistemin Əsası: Reqressiya testi proqramın kod bazasında edilən dəyişiklikləri izləmək üçün versiyaya nəzarət sistemlərinə əsaslanır. Avtomatlaşdırılmış Reqressiya Testi: Avtomatlaşdırma alətləri reqressiya testlərinin icrasını asanlaşdırmaq və sürətləndirmək üçün istifadə edilə bilər. Təsirin Təhlili: Reqressiya testi dəyişikliklərin mövcud funksiyalara təsirinin təhlilini əhatə edir. Qüsurların İdarə Olunması: Reqressiya testi dəyişikliklərin yaratdığı hər hansı qüsurları və ya problemləri müəyyən edir. Davamlı Reqressiya Testi: Çevik inkişaf mühitlərində və ya tətbiqdə tez-tez dəyişikliklər edildikdə, davamlı reqressiya testi istifadə olunur.

Reqressiya testi dəyişikliklərdən, xətaların aradan qaldırılmasından və ya yeni funksiyaların tətbiqindən sonra veb tətbiqinin gözlənildiyi kimi işləməyə davam etməsini təmin edir.

7.Yük testi: Yük testi belə şərtlərdə veb tətbiqinin performansını və davranışını qiymətləndirmək üçün yüksək istifadəçi yüklərini simulyasiya edir.

Yük testi, gözlənilən istifadəçi yükləri altında tətbiqin performansını və davranışını qiymətləndirməyə yönəlmiş bir veb tətbiqi test növüdür. Yük testinin əsas aspektləri bunlardır:

Test Ssenariləri və İş Yükləri: Yük testi istifadəçinin veb tətbiqi ilə qarşılıqlı əlaqəsini simulyasiya edən real sınaq ssenarilərinin yaradılmasını əhatə edir. Performans Metrikləri: Yük testi yük altında tətbiqin performansını qiymətləndirmək üçün müxtəlif performans ölçülərini ölçür. Test Mühitinin Quraşdırılması: Yük testi aparat, proqram təminatı və şəbəkə konfiqurasiyaları baxımından istehsal mühitinə

yaxından bənzəyən sınaq mühiti tələb edir. Testin icrası və monitorinqi: Yük testi müəyyən edilmiş iş yükü ilə müəyyən edilmiş test ssenarilərinin icrasını və tətbiqin işinə nəzarəti əhatə edir. Ölçüləbilənlik Testi: Yük testinə tətbiqin artan trafik səviyyələrini necə ölçdüyünü və idarə etdiyini qiymətləndirmək üçün istifadəçi yükünü tədricən artırmaqla veb tətbiqinin miqyaslılığının qiymətləndirilməsi daxildir. Stress testi: Stress testi, həddindən artıq yük şəraitində davranışını qiymətləndirmək üçün tətbiqi gözlənilən imkanlarından kənara itələməyi əhatə edən yük testinin bir variantıdır. Performans Tənzimləmə: Yük testi performans problemlərini və optimallaşdırma sahələrini müəyyən etməyə kömək edir. Hesabat və Təhlil: Performans problemlərini, mənfiləri və təkmilləşdirilməli sahələri müəyyən etmək üçün yük testinin nəticələri təhlil edilir.

Yük testi keçirməklə təşkilatlar veb tətbiqinin gözlənilən istifadəçi yükünü idarə edə bilməsini, məqbul cavab vaxtlarını saxlamasını və qənaətbəxş istifadəçi təcrübəsi təqdim edə biləcəyini təmin edə bilər.

8.Stress Testi: Stress testi veb tətbiqinin artıq yükləmələr, yüksək trafik, məhdud resurslar və ya istifadədən imtinalar kimi ekstremal və ya anormal şərtləri idarə etmək qabiliyyətini qiymətləndirir. Stress testi, həddindən artıq və ya normal yüklənmə şəraitində tətbiqin davranışını və performansını qiymətləndirməyə yönəlmiş veb tətbiqi testinin bir növüdür. Stress testinin əsas aspektləri bunlardır:

Test Ssenariləri: Stress testi qeyri-adi yüksək istifadəçi yüklərini və ya stress şərtlərini simulyasiya edən sınaq ssenarilərinin müəyyən edilməsini və icrasını əhatə edir. Yük yaratma: Tətbiqdə stress tətbiq etmək üçün stress testi tətbiqin normal işləmə qabiliyyətindən əhəmiyyətli dərəcədə artıq yük yaratmağı nəzərdə tutur. Performans Monitorinqi: Stress testi artan yük altında tətbiqin performansının və davranışının monitorinqini əhatə edir. Qırılma Nöqtəsinin İdentifikasiyası: Stress testi tətbiqin qırılma nöqtəsini və ya maksimum yükləmə qabiliyyətini təyin etmək məqsədi daşıyır. Uğursuzluğun Bərpası və Dayanıqlılığı: Stress testi tətbiqin stresdən qaynaqlanan uğursuzluqlardan və ya anormal şəraitdən necə bərpa olunduğunu müşahidə edir. Resursdan İstifadə və Ölçüləbilmə: Stress testi həddindən artıq yük şəraitində tətbiqin CPU, yaddaş, disk giriş/çıxışı və şəbəkə genişliyi kimi resurs istifadəsini

qiymətləndirir. Səhvlərin idarə edilməsi və qeyd: Stress testinə tətbiqin səhvlərin idarə edilməsi mexanizmlərinin və xəta qeydinin adekvatlığının qiymətləndirilməsi daxildir. Performansın bərpası və optimallaşdırılması: Stress testindən sonra tətbiqin yüksək yükləri idarə etmək qabiliyyətini yaxşılaşdırmaq üçün performansın bərpası tədbirləri və optimallaşdırma üsulları həyata keçirilə bilər.

Stress testi ekstremal şəraitdə tətbiqin möhkəmliyini, dayanıqlığını və sabitliyini qiymətləndirmək üçün vacibdir.

9.Əlçatanlıq Testi: Əlçatanlıq testi veb tətbiqinin əlilliyi olan istifadəçilər üçün əlçatan olmasını təmin etməyə yönəlmişdir. Əlçatanlıq testi, tətbiqin əlilliyi olan şəxslər üçün əlçatanlığının və əlyetərliyinin qiymətləndirilməsinə yönəlmiş veb tətbiqi testinin bir növüdür. Əlçatanlıq testinin əsas aspektləri bunlardır:

Əlçatanlıq Standartlarına Uyğunluq: Əlçatanlıq testi veb tətbiqinin Ümumdünya Veb Konsorsiumu (W3C) tərəfindən nəşr olunan Veb Məzmununa Əlçatanlıq Təlimatları (WCAG) kimi əlçatanlıq standartlarına və təlimatlara uyğunluğunun yoxlanılmasını əhatə edir. Klaviatura naviqasiyası: Əlçatanlıq testi veb proqramın bütün funksiyalarına maus və ya toxunuşla qarşılıqlı əlaqəyə etibar etmədən yalnız klaviaturadan istifadə etməklə daxil olmaq və idarə oluna biləcəyini təmin edir. Ekran Oxucu Uyğunluğu: Əlçatanlıq testi veb tətbiqinin ekran oxuyucu proqramı ilə uyğunluğunun yoxlanılmasını nəzərdə tutur. Şəkillər üçün Alternativ Mətn: Əlçatanlıq testi veb proqramdakı şəkillərin müvafiq alternativ mətn təsvirlərinə malik olduğunu yoxlayır. Rəng Kontrastı və Vizual Elementlər: Əlçatanlıq testi görmə qüsuru olan şəxslər üçün oxunaqlılığı təmin etmək üçün mətn və fon rəngləri arasında rəng kontrast nisbətini qiymətləndirir. Video və Audio Əlçatanlıq: Əlçatanlıq testi veb proqramdakı video və audio məzmunun başlıqlar, transkriptlər və ya audio təsvirləri ilə müşayiət olunmasını təmin edir. Forma və Daxiletmə Doğrulaması: Əlçatanlıq testi forma sahələrinin müvafiq etiketlərə, təlimatlara və səhv mesajlarına malik olduğunu yoxlayır. Köməkçi Texnologiya Uyğunluğu: Əlçatanlıq testi veb tətbiqinin ekran oxuyucuları, böyüdücülər, səs tanıma proqramı və ya keçid cihazları kimi müxtəlif köməkçi texnologiyalarla uyğunluğunun yoxlanılmasını nəzərdə tutur.

10.Lokallaşdırma Testi: Lokallaşdırma testi veb tətbiqinin müxtəlif dillərdə, bölgələrdə və mədəniyyətlərdə uyğunlaşdırılmasını və düzgün işləməsini təmin edir. Lokallaşdırma testi xüsusi hədəf yerli və ya regionlar üçün tətbiqin funksionallığını, istifadəyə yararlılığını və linqvistik uyğunlaşma qabiliyyətini yoxlamağa yönəlmiş veb tətbiqi testinin bir növüdür. Lokalizasiya testinin əsas aspektləri bunlardır:

Dil dəstəyi: Lokallaşdırma testi veb tətbiqinin hədəf auditoriyanın istədiyiniz dillərini dəstəklədiyini yoxlayır. Tarix və Saat Formatları: Lokallaşdırma testi veb tətbiqinin hədəf bölgəyə xas olan tarix və vaxt formatlarını düzgün idarə etdiyini yoxlamaqdan ibarətdir. Valyuta və Rəqəm Formatları: Lokallaşdırma sınağı veb tətbiqinin valyuta simvollarını, onluq ayırıcıları, min ayırıcıları və hədəf lokalda istifadə olunan rəqəmsal formatları düzgün idarə etməsini təmin edir. Mədəni Uyğunlaşma: Buraya mədəni cəhətdən həssas elementlərin, simvolların, ikonaların, rənglərin, təsvirlərin və yerli adət və ya ənənələrin müvafiq istifadəsinin yoxlanılması daxildir. Yerli Qaydalar və Hüquqi Tələblər: Lokallaşdırma testi veb tətbiqinin yerli qaydalara və hədəf bölgəyə xas olan qanuni tələblərə uyğunluğunu yoxlamağı əhatə edir. İstifadəçi interfeysinin uyğunlaşdırılması: Lokallaşdırma testi istifadəçi interfeysi elementlərinin, məsələn, düymələr, menyular, formalar və etiketlərin düzgün tənzimləndiyini və lokallaşdırılmış mətnə uyğun olduğunu yoxlayır. İstifadəçi Girişinin Təsdiqlənməsi: Lokallaşdırma sınağına müxtəlif dillər və simvol dəstləri üçün veb tətbiqinin daxiletmənin yoxlanılması və xətaların idarə edilməsi mexanizmlərinin sınaqdan keçirilməsi daxildir. Lokallaşdırma Ardıcıllığı: Lokallaşdırma testi bütün istifadəçi interfeysi elementləri və məzmununda vahid tərcümə, terminologiya və dil istifadəsini yoxlayaraq bütün tətbiqdə ardıcıllığı təmin edir. O, yerli versiyanın mənbə dil və ya istinad materialları ilə müqayisədə ardıcıllığı və dəqiqliyi saxladığını yoxlayır.

Hərtərəfli lokallaşdırma testini həyata keçirməklə təşkilatlar veb tətbiqinin hədəf auditoriyanın dil və mədəniyyət üstünlüklərinə effektiv şəkildə cavab verməsini təmin edə bilərlər. Lokallaşdırma testi istifadəçinin qəbuluna mane ola biləcək və ya istifadəçi təcrübəsinə mənfi təsir göstərə biləcək hər hansı lokalizasiya problemlərini, dillə bağlı səhvləri və ya mədəni uyğunsuzluqları müəyyən etməyə və düzəltməyə

kömək edir. O, veb tətbiqinə yerli auditoriya ilə rezonans yaratmağa, istifadəçi məmnuniyyətini artırmağa və qlobal bazarlarda rəqabət üstünlüyü əldə etməyə imkan verir.

Veb tətbiqi sınağı üçün adətən testin planlaşdırılması, test işinin dizaynı, icrası, qüsurların hesabatı və izlənilməsi ilə başlayan strukturlaşdırılmış yanaşma əsasında aparılır.

Sınaq prosesini avtomatlaşdırmaq və sadələşdirmək, səmərəliliyi və effektivliyi artırmaq üçün müxtəlif sınaq üsulları, alətlər və çərçivələrdən istifadə olunur.

1.2. QA Avtomatlaşdırılmasına giriş

Avtomatlaşdırılmış sınaq kimi də tanınan QA avtomatlaşdırılması, proqram təminatı və ya sistemində əvvəlcədən hazırlanmış sınaqları yerinə yetirmək üçün proqram alətlərindən istifadə prosesidir. QA avtomatlaşdırılması və ya Keyfiyyət Təminatının avtomatlaşdırılması, proqram təminatının sınaqdan keçirilməsini avtomatlaşdırmaq üçün proqram alətləri və skriptlərdən istifadə təcrübəsidir. [1] QA avtomatlaşdırılmasının məqsədi proqram təminatının sınaqdan keçirilməsi prosesinin səmərəliliyini və effektivliyini artırmaq və əllə sınaq üçün tələb olunan vaxt və səyləri azaltmaqdır. Proqram təminatının hazırlanmasında sınaq proqram təminatının müəyyən edilmiş tələblərə cavab verməsini, qüsur və səhvlərdən azad olmasını təmin edən kritik mərhələdir. Manual testlər vaxt aparan, təkrarlanan və insan səhvinə meylli ola bilər. QA avtomatlaşdırılması bir çox test tapşırıqlarını avtomatlaşdıraraq və onları daha sürətli, daha dəqiq və etibarlı etməklə həll edir. [23]

Ümumilikdə, QA avtomatlaşdırılması müasir proqram təminatının inkişafının vacib hissəsidir və komandalara yüksək keyfiyyətli proqram təminatını daha sürətli templə və məhsullarının etibarlılığına və sabitliyinə daha çox inamla çatdırmağa imkan verir.

QA avtomatlaşdırılması bir neçə səbəbə görə vacibdir, üstünlükləri:

Təkmilləşdirilmiş effektivlik, Ardıcıl və etibarlı test, Daha geniş sınaq əhatə dairəsi, Qüsurların erkən aşkarlanması, Xərc baxımından səmərəli, xərclərə qənaət,

Daha yaxşı əməkdaşlıq, Artan Səmərəlilik, Təkmilləşdirilmiş Dəqiqlik, Daha yüksək sınaq əhatə dairəsi, Daha keyfiyyətli proqram təminatı.

Təkmilləşdirilmiş effektivlik: Avtomatlaşdırma testləri tez və təkrar-təkrar icra etməyə imkan verir, QA komandası üzvlərinin diqqətini digər tapşırıqlara yönəltmək üçün vaxtını azad edir. Bu, daha səmərəli sınaq prosesinə, daha sürətli buraxılışlara və innovasiya və optimallaşdırma üçün daha çox vaxta səbəb ola bilər.

Ardıcıl və etibarlı test: Avtomatlaşdırılmış testlər hər dəfə eyni şəkildə həyata keçirilir, yəni nəticələr ardıcıl və etibarlıdır. Bu, insan səhvi riskini azaltmağa kömək edir və sınaq prosesinin dəqiq və təkrarlanmasını təmin edir.

Daha geniş sınaq əhatə dairəsi: Avtomatlaşdırma test qruplarına yalnız əllə sınaqdan keçirməklə mümkün olduğundan daha geniş diapazonlu test işləri yaratmağa və icra etməyə imkan verir. Bu, testin əhatə dairəsini yaxşılaşdırmağa və proqram tətbiqində səhv və problem riskini azaltmağa kömək edir.

Qüsurların erkən aşkarlanması: Avtomatlaşdırılmış sınaq qüsurları və problemləri inkişaf dövrünün əvvəlində müəyyən etməyə kömək edə bilər ki, bu da onları düzəltməyi asanlaşdırır və daha ucuz edir. Bu, son məhsulda daha az qüsurlara və daha yaxşı ümumi keyfiyyətə səbəb ola bilər.

Xərc baxımından səmərəli: Avtomatlaşdırmaya ilkin investisiya yüksək ola bilsə də, çox vaxt uzun müddətdə daha sərfəli olur. Avtomatlaşdırılmış testlər əlavə resurs tələb etmədən dəfələrlə həyata keçirilə bilər ki, bu da testin ümumi dəyərini azaltmağa kömək edir.

Daha yaxşı əməkdaşlıq: Avtomatlaşdırma inkişaf və sınaq qrupları arasında əməkdaşlığı yaxşılaşdırmağa kömək edə bilər. Test tapşırıqlarını avtomatlaşdırmaqla inkişaf qrupları daha tez rəy əldə edə və sınaq qrupları ilə daha effektiv əməkdaşlıq edə bilər.

Bu üstünlüklərə əlavə olaraq, QA avtomatlaşdırılması həm də inkişaf və sınaq qrupları arasında əməkdaşlığı yaxşılaşdırmağa, daha dəqiq hesabat və ölçüləri təmin etməyə və resurslardan daha səmərəli istifadə etməyə imkan verə bilər. Ümumilikdə, QA avtomatlaşdırılması proqram təminatının yoxlanılması prosesinin səmərəliliyini, effektivliyini və düzgünlüyünü təkmilləşdirmək istəyən proqram təminatı inkişaf

qrupları üçün vacib vasitədir və daha sürətli və sərfəli şəkildə çatdırılan yüksək keyfiyyətli proqram təminatıdır.

Həmçinin manual testinin avtomatlaşdırılmış testlə müqayisədə avtomatlaşdırmadan istifadənin üstünlükləri var.

QA avtomatlaşdırılmasının çətinlikləri və mənfi cəhətləri:

QA avtomatlaşdırılmasının bir çox üstünlükləri olsa da, təşkilatların avtomatlaşdırmanı həyata keçirərkən qarşılaşdıqları bəzi çətinliklər də var. Ən çox görülən çətinliklərdən bəziləri bunlardır:

Alət seçimi, alətlərdən asılılıq: Avtomatlaşdırılmış sınaq alətlər və proqram təminatından asılıdır və alətlər və ya mühitlə bağlı hər hansı problem sınaq prosesinə təsir göstərə bilər.

Test işi seçimi, test baxımı: Avtomatlaşdırılmış testlərin aparılması çox vaxt aparan və mürəkkəb ola bilər.

Texniki problemlər: Avtomatlaşdırma bəzən obyektin tanınması ilə bağlı çətinliklər, sinxronizasiya problemləri və ya tətbiqin UI-də gözlənilməz dəyişikliklər kimi texniki problemlərlə üzləşə bilər.

Xərc, Yüksək İlkin Xərc: Avtomatlaşdırılmış sınaq çərçivəsinin qurulmasının ilkin dəyəri alətlərin, lisenziyaların və avadanlıqların qiyməti daxil olmaqla yüksək ola bilər.

Bacarıqların və təcrübənin olmaması, Bacarıqlı Sınaqçılara Ehtiyac: Avtomatlaşdırılmış sınaq üçün tapmaq və saxlamaq çətin ola bilən avtomatlaşdırılmış test skriptlərinin yazılması və saxlanması təcrübəsi olan təcrübəli sınaqçılar tələb olunur.

Maintenance Overhead: Avtomatlaşdırılmış test skriptləri tətbiq dəyişdikcə texniki xidmət, yeniləmələr və dəyişikliklər tələb edir ki, bu da sınaq prosesinə əlavə məsrəflər və xərclər əlavə edə bilər.

Məhdud əhatə dairəsi: Avtomatlaşdırılmış sınaq bütün sınaq növləri üçün uyğun deyil və əllə testin daha effektiv və ya zəruri olduğu ssenarilər ola bilər.

Yanlış təhlükəsizlik hissi: Avtomatlaşdırılmış sınaq bütün sınaqların əhatə olunduğunu fərz etməklə yanlış təhlükəsizlik hissi bilər, halbuki bəzi ssenarilərin

sınaqdan keçirilməməsi və potensial olaraq istehsalda problemlər yarada bilməsi mümkündür.

Yaradıcılıq və İnsightın olmaması: Avtomatlaşdırılmış testlər əl ilə testin yaradıcılığı və anlayışından məhrum ola bilər, burada testçilər tətbiqi araşdıra və avtomatlaşdırılmış testlərlə əhatə olunmayan potensial problemləri müəyyən edə bilərlər. [22]

Avtomatlaşdırma testinin üstünlüklərini və mənfi cəhətlərini diqqətlə nəzərdən keçirmək və layihənin xüsusi tələblərinə və məqsədlərinə əsaslanaraq uyğun yanaşma seçmək vacibdir.

1.3. Avtomatik testetmə ilə Manual testetmə

Şübhəsiz ki, avtomatlaşdırılmış sınaq ilə əl testinin üstünlüklərini müqayisə edək və avtomatlaşdırmadan istifadənin üstünlüklərini vurğulayaq:

Sürət və Səmərəlilik: Avtomatlaşdırılmış sınaq əllə sınaqdan daha sürətli və daha səmərəlidir. Avtomatlaşdırılmış testlər təkrar-təkrar və daha sürətli icra oluna bilər, əllə sınaq isə insan testerlərindən sınaq prosesinin hər bir addımını yerinə yetirməsini tələb edir, bu da vaxt aparan və səhvlərə meyilli ola bilər.

Testin əhatə dairəsi: Avtomatlaşdırılmış sınaq əl ilə sınaqdan daha çox sınaq əhatəsini təmin edə bilər, çünki kənar hallar və mənfi ssenarilər də daxil olmaqla daha geniş ssenari və şərtləri sınamaq mümkündür.

Dəqiqlik və Etibarlılıq: Avtomatlaşdırılmış sınaq əllə sınaqdan daha dəqiq və etibarlıdır. Avtomatlaşdırılmış testlər hər dəfə eyni şəkildə həyata keçirilir və insan xətası üçün daha az yer var.

Xərc-effektivlik: Əllə sınaqdan keçirilməsi insan resurslarını tələb etsə də, avtomatlaşdırılmış sınaq uzunmüddətli perspektivdə daha sərfəli ola bilər.

Əməkdaşlıq: Avtomatlaşdırılmış sınaq inkişaf və sınaq qrupları arasında daha yaxşı əməkdaşlığa imkan verir.

Yenidən istifadə oluna bilməsi: Avtomatlaşdırılmış testlər təkrar-təkrar istifadə oluna bilər, halbuki hər dəfə əllə sınaqlar yaradılmalı və icra edilməlidir, bu da onu daha çox vaxt aparır.

Ümumilikdə, əllə testin insan intuisiyasından və yaradıcılığından istifadə etmək bacarığı kimi bəzi üstünlükləri ola bilsə də, avtomatlaşdırılmış sınaq sürət, səmərəlilik, sınaq əhatə dairəsi, dəqiqlik, etibarlılıq, iqtisadi səmərəlilik, əməkdaşlıq və təkrar istifadə imkanları da daxil olmaqla əhəmiyyətli üstünlüklər təklif edir. Buna görə də, avtomatlaşdırılmış test proqram təminatının inkişaf etdirilməsi qrupları üçün sınaq prosesinin səmərəliliyini, effektivliyini və dəqiqliyini artırmaq üçün zəruri bir vasitədir və daha sürətli və sərfəli şəkildə çatdırılan yüksək keyfiyyətli proqram təminatına gətirib çıxarır.Avtomat və əl testi arasında fərq:

Avtomatlaşdırma və əl testi arasındakı əsas fərq test prosesinə insan testerlərinin cəlb edilməsidir. Əl testində insan test cihazı test işlərini əl ilə yerinə yetirir, [20] Avtomatlaşdırma testində isə test işləri avtomatlaşdırma alətləri və skriptlərdən istifadə etməklə həyata keçirilir. Avtomatlaşdırma və əl testi arasındakı digər fərqlər bunlardır:

Sürət və Səmərəlilik: Avtomatlaşdırma sınağı ümumiyyətlə manual testdən daha sürətli və daha səmərəlidir, çünki o, çox sayda test işini tez və dəqiq şəkildə yerinə yetirə bilər. [20]

Testin əhatə dairəsi: Avtomatlaşdırma testi daha çox sınaq əhatəsini təmin edə bilər, çünki o, çoxlu sayda test işini yerinə yetirə bilər, halbuki əl ilə sınaq vaxt və resurs məhdudiyyətlərinə görə bütün mümkün ssenariləri əhatə edə bilməz.

Baxım: Avtomatlaşdırılmış test skriptləri tətbiq dəyişdikcə texniki qulluq, yeniləmələr və dəyişikliklər tələb edir, halbuki əl ilə sınaq işləri daha asan yenilənə və dəyişdirilə bilər. Bu, sınaq prosesinə əlavə məsrəflər və xərclər əlavə edə bilər.

İnsan xətası: Manual sınaq insan səhvinə daha həssasdır, çünki o, testləri dəqiq yerinə yetirmək üçün insan sınayıcılarına əsaslanır, halbuki avtomatlaşdırma testi əvvəlcədən təyin edilmiş test skriptlərindən istifadə etdiyi üçün insan səhvinə daha az meyllidir.

Xərc: Avtomatlaşdırılmış sınaq çərçivəsinin qurulmasının ilkin dəyəri alətlər, lisenziyalar və avadanlıqların qiyməti daxil olmaqla yüksək ola bilər.

Yaradıcılıq və İnsight: Manual test daha çox yaradıcılıq və fikir əldə etməyə imkan verir, çünki insan sınayıcıları tətbiqi araşdıra və avtomatlaşdırılmış testlərlə əhatə olunmayan potensial problemləri müəyyən edə bilər.

Avtomatlaşdırmanın və əl ilə sınaqların üstünlüklərini və mənfi cəhətlərini diqqətlə nəzərdən keçirmək və layihənin xüsusi tələblərinə və məqsədlərinə əsaslanaraq uyğun yanaşma seçmək vacibdir. Ümumiyyətlə, həm əllə, həm də avtomatlaşdırma testinin birləşməsi test əhatə dairəsi, sürət və səmərəlilik baxımından ən yaxşı nəticələri təmin edə bilər. [20]

Hansı istifadəyə yararlıdır? - Həm avtomatlaşdırma, həm də əl testi öz güclü və zəif tərəflərinə malikdir və hansının daha çox istifadəyə yararlı olması layihənin xüsusi tələblərindən və məqsədlərindən asılıdır.

1.4. Selenium veb test haqqında tədqiqat məqsədi

Selenium veb testi haqqında tezisin məqsədi müəllifin xüsusi tədqiqat məqsəd və məqsədlərindən asılı olaraq dəyişə bilər. Bununla birlikdə, Selenium veb testi ilə bağlı tezisin bəzi ümumi məqsədləri aşağıdakıları əhatə edə bilər:

Seleniumun bir sınaq vasitəsi kimi qiymətləndirilməsi: Tezis Seleniumun imkanlarını, xüsusiyyətlərini və məhdudiyyətlərini veb test aləti kimi qiymətləndirmək məqsədi daşıya bilər. O, Selenium VebDriver, Selenium Grid və Selenium IDE kimi Seleniumun müxtəlif komponentlərini tədqiq edə və veb tətbiqi testlərinin avtomatlaşdırılmasında onların effektivliyini qiymətləndirə bilər. [4]

Digər Test Alətləri ilə Müqayisə: Tezis Seleniumun üstünlüklərini, çatışmazlıqlarını və müxtəlif sınaq ssenariləri üçün uyğunluğunu müəyyən etmək üçün digər məşhur veb test alətləri ilə müqayisə edə bilər.

Avtomatlaşdırma Çərçivəsinin İnkişafı: Dissertasiya xüsusi sınaq ehtiyaclarını və ya problemlərini həll etmək üçün Selenium istifadə edərək avtomatlaşdırma çərçivəsini inkişaf etdirməyə yönəldilə bilər.

Test Skriptinin Dizaynı və Optimallaşdırılması: Tezis Seleniumdan istifadə edərək effektiv və səmərəli test skriptlərinin dizaynı üçün ən yaxşı təcrübələri araşdıra bilər.

Davamlı İnteqrasiya/Davamlı Çatdırılma (CI/CD) ilə İnteqrasiya: Tezis proqram təminatının hazırlanması və yerləşdirilməsi prosesinin bir hissəsi kimi problemsiz və

avtomatlaşdırılmış veb tətbiqi testini təmin etmək üçün Seleniumun CI/CD ilə inteqrasiyasını araşdıra bilər.

Selenium ilə Performans Testi: Selenium-dan istifadə edərək veb tətbiqinin performansını ölçmək və təhlil etmək və onun yük testi, stress testi və genişlənmə testi üçün uyğunluğunu qiymətləndirmək üçün üsulları araşdıra bilər.

Test Baxımı və Sağlamlıq: O, veb proqramları inkişaf etdikcə test skriptlərinin saxlanması, UI elementlərində dəyişikliklərin idarə edilməsi və avtomatlaşdırılmış testlərin etibarlılığını və dayanıqlığını artırmaq strategiyalarını araşdıra bilər.

Case Studies və Empirik Qiymətləndirmələr: Bu, xüsusi kontekstlərdə Seleniumun effektivliyini və səmərəliliyini təsdiqləmək üçün təcrübələr aparmaq, məlumat toplamaq və nəticələrin təhlilini əhatə edə bilər.

Ümumiyyətlə, Selenium veb sınağı haqqında tədqiqatın məqsədi Seleniumun veb sınaq vasitəsi kimi praktiki istifadəsi, çətinlikləri və potensial təkmilləşdirmələri haqqında araşdırmaq, təhlil etmək və məlumat verməklə mövcud biliklər toplusuna töhfə verməkdir. Tədqiqat veb tətbiqi testləri sahəsini inkişaf etdirmək, praktikantlar üçün tövsiyələr vermək və ya avtomatlaşdırılmış test üsullarının akademik anlayışına töhfə vermək məqsədi daşıya bilər.

II FƏSİL. SELENİUM VEB TESTETMƏ

Proqram təminatının sınaqdan keçirilməsi üçün bir neçə test avtomatlaşdırma çərçivəsi mövcuddur və ən populyarlarından bəziləri Selenium, Appium və TestComplete-dir.

Selenium, veb proqramların sınaqdan keçirilməsi üçün geniş istifadə olunan açıq mənbəli test avtomatlaşdırma çərçivəsidir. Java, Python və C# kimi bir çox proqramlaşdırma dillərini dəstəkləyir və bir çox brauzer və platformada işləyə bilər. Selenium test edənlərə funksional, reqressiya və performans testləri daxil olmaqla veb proqramlar üçün avtomatlaşdırılmış testlər yaratmağa və icra etməyə imkan verir. Çərçivə veb tətbiqi ilə qarşılıqlı əlaqə yaratmaq və istifadəçi hərəkətlərini simulyasiya etmək üçün veb sürücülərin və API-lərin birləşməsindən istifadə edir. [13]

Selenium test edənlərə müxtəlif brauzerlər və platformalarda veb-əsaslı proqramları avtomatlaşdırmağa imkan verən sınaq avtomatlaşdırma çərçivəsidir. Selenium VebDriver testerlərə düymələr, daxiletmə sahələri və açılan menyular kimi veb elementləri ilə qarşılıqlı əlaqə yaratmağa imkan verən bir sıra API təmin edir. Selenium Java, Python və C# kimi bir çox proqramlaşdırma dillərini dəstəkləyir və sınaqçılara avtomatlaşdırılmış testləri seçdikləri dildə yazmağa imkan verir. Seleniumun əsas xüsusiyyətlərindən bəziləri bunlardır:

Cross brauzer sınağı: Selenium Chrome, Firefox və Safari kimi müxtəlif brauzerləri dəstəkləyir, bu da test edənlərə brauzerlər arası sınaq keçirməyə imkan verir.

Fərqli platformalar üçün dəstək: Selenium Windows, Mac və Linux kimi bir çox platformanı dəstəkləyir, bu da testçilərə platformalararası sınaq keçirməyə imkan verir. Çoxlu proqramlaşdırma dilləri üçün dəstək: Selenium müxtəlif proqramlaşdırma dillərini dəstəkləyir, bu da test edənlərin üstünlük verdiyi dildə avtomatlaşdırılmış testlər yazmasını asanlaşdırır. Test çərçivələri ilə asan inteqrasiya: Selenium asanlıqla TestNG və JUnit kimi sınaq çərçivələri ilə inteqrasiya oluna bilər.

Selenium, veb proqramların funksional və reqressiya testlərini yerinə yetirmək üçün veb brauzerləri avtomatlaşdırmaq üçün istifadə olunan geniş istifadə olunan açıq

mənbəli vasitədir. O, Selenium IDE, Selenium VebDriver və Selenium Grid daxil olmaqla veb brauzerləri avtomatlaşdırmaq üçün birlikdə işləyən çoxsaylı komponentlərdən ibarətdir. Selenium IDE, sınaqçılara istifadəçi hərəkətlərini qeyd etməyə və avtomatik olaraq test skriptləri yaratmağa imkan verən qeyd və oxutma vasitəsidir. Selenium VebDriver, test edənlərə Java, Python, C# və Ruby kimi müxtəlif proqramlaşdırma dillərində daha mürəkkəb test skriptləri yazmağa imkan verən proqramlaşdırma interfeysidir. Selenium Grid test edənlərə test skriptlərini paralel olaraq birdən çox maşın və brauzerdə işlətməyə imkan verən və daha sürətli sınaq və daha geniş sınaq əhatəsinə imkan verən bir vasitədir. Bununla belə, veb proqramları sınaqdan keçirmək üçün Selenium istifadəsinin bəzi potensial çatışmazlıqlarına proqramlaşdırma bacarıqlarına ehtiyac, texniki xidmət xərcləri və brauzer uyğunluğu problemləri daxildir. Ümumilikdə, Selenium veb proqram testlərini avtomatlaşdırmaq üçün güclü bir vasitədir və sənayedə geniş istifadə olunur. Seleniumun üstünlüklərini və mənfi cəhətlərini diqqətlə qiymətləndirmək və layihənin xüsusi tələblərinə və məqsədlərinə əsaslanaraq uyğun yanaşma seçmək vacibdir.

Niyə Selenium? Seleniumun bazardakı digər veb avtomatlaşdırma vasitələri ilə müqayisədə daha populyar və güclü olmasının bir çox səbəbi var. Bunun səbəbləri bunlardır: Bu, açıq mənbəli, istifadəsi pulsuz, portativ bir vasitədir və veb-testi qüsursuz şəkildə həyata keçirmək üçün istifadə olunur. Bu, müxtəlif növ testləri həyata keçirməyə imkan verən bir neçə Domain Spesifik Dil (DSL) və digər alətlərin birləşməsidir. Anlamaq olduqca asandır, intuitivdir və aşağı öyrənmə əyrisinə malikdir. Əmrlər çox siniflər kimi təsnif edilir, bu da onları başa düşməyi asanlaşdırır. Xüsusilə təkrar sınaq vəziyyətlərində testlərin aparılması üçün tələb olunan vaxt kəskin şəkildə azaldığından, sınaqçılardan yükü götürür. Biznes müştərilərinə çəkilən xərclərdə əhəmiyyətli azalma var ki, bu da qazan-qazan vəziyyətidir.

2.1. Selenium veb test etmədən istifadə

Selenium, veb proqramlarını sınaqdan keçirmək üçün istifadə edilən açıq mənbəli avtomatlaşdırılmış test vasitəsidir. O, linklərə klikləmək, formaları doldurmaq və gözlənilən nəticələri təsdiqləmək kimi brauzer hərəkətlərini avtomatlaşdırmağın bir

yolunu təqdim edir. Selenium Java, Python, C#, Ruby və başqaları kimi bir çox proqramlaşdırma dillərini dəstəkləyir. Selenium VebDriver-dən istifadə etməyə başlamaq üçün əvvəlcə sınaqdan keçirmək istədiyiniz veb brauzer üçün tələb olunan sürücüləri quraşdırmalısınız. Sürücüləri quraşdırdıqdan sonra Selenium VebDriver API-dən istifadə edərək test skriptlərinizi yaza bilərsiniz. API veb elementləri tapmaq, onlarla qarşılıqlı əlaqə qurmaq və onların xassələrini yoxlamaq üçün üsullar təqdim edir.

Ümumilikdə, Selenium, təkrarlanan tapşırıqları avtomatlaşdırmağa, vaxta qənaət etməyə və veb tətbiqlərinizin keyfiyyətini yaxşılaşdırmağa kömək edə bilən veb testi üçün güclü bir vasitədir.

Üstünlüklərini qeyd etmək lazımdır:

Açıq mənbə: Selenium açıq mənbəli bir vasitədir, yəni istifadə etmək pulsuzdur və heç bir lisenziya xərcləri yoxdur. Brauzerlər arası uyğunluq: Selenium Chrome, Firefox, Safari, Edge və Internet Explorer daxil olmaqla bir çox veb brauzerləri dəstəkləyir. Bu o deməkdir ki, siz veb tətbiqinizi müxtəlif brauzerlərdə sınaqdan keçirə və onların hamısında gözlənildiyi kimi işləməsini təmin edə bilərsiniz. Dil dəstəyi: Selenium Java, Python, C#, Ruby və JavaScript daxil olmaqla bir çox proqramlaşdırma dillərini dəstəkləyir. Çeviklik: Selenium testlərinizi necə yazmağınız və icra etdiyiniz baxımından çox rahatlıq təmin edir. İnteqrasiya: Selenium JUnit, TestNG və NUnit kimi məşhur sınaq çərçivələri ilə inteqrasiya edə bilər. Yenidən istifadə edilə bilər: Selenium testləri inkişaf, sınaq və istehsal kimi müxtəlif mühitlərdə təkrar istifadə edilə bilər. Ümumilikdə, Selenium veb tətbiqlərinizin keyfiyyətini və etibarlılığını təmin etməyə kömək edə bilən veb testi üçün güclü və çevik bir vasitədir.

Mənfi cəhətlərini qeyd etmək lazımdır:

Selenium veb testi üçün güclü və populyar bir vasitə olsa da, onun bəzi çatışmazlıqları var ki, bunlardan xəbərdar olmaq lazımdır:

Daxili hesabatın olmaması: Seleniumun daxili hesabat imkanları yoxdur, bu o deməkdir ki, hesabat yaratmaq və test nəticələrini təhlil etmək üçün üçüncü tərəf alətlərindən istifadə etməli və ya xüsusi kod yazmalısınız.

Vaxt aparan quraşdırma: Seleniumun qurulması çox vaxt apara bilər, xüsusən də birdən çox brauzer və əməliyyat sistemi üçün sürücüləri və mühitləri konfiqurasiya etməlisiniz.

Qeyri-veb texnologiyaları üçün məhdud dəstək: Selenium ilk növbədə veb proqramları sınaqdan keçirmək üçün nəzərdə tutulub, bu o deməkdir ki, o, masaüstü proqramlar və ya mobil proqramlar kimi qeyri-veb texnologiyalarını sınaqdan keçirmək üçün ən yaxşı seçim olmaya bilər.

Vizual test üçün məhdud dəstək: Selenium vizual test üçün daxili dəstəyi təmin etmir, yəni vizual reqressiya testini yerinə yetirmək üçün üçüncü tərəf alətlərindən istifadə etməli və ya xüsusi kod yazmalısınız.

Ümumiyyətlə, Selenium veb testi üçün güclü bir vasitə olsa da, onun sınaq ehtiyaclarınız üçün istifadə edib-etməməsinə qərar verərkən nəzərə almalı olduğunuz bəzi məhdudiyyətlər və problemlər var.

2.2. Selenium nədir və necə işləyir?

Selenium, veb brauzerlərin avtomatlaşdırılması üçün geniş istifadə olunan açıq mənbəli çərçivədir. O, veb proqram testini avtomatlaşdırmaq, formaları doldurmaq, düymələri basmaq və veb səhifələrdə naviqasiya kimi hərəkətləri yerinə yetirmək üçün bir sıra alətlər və kitabxanalar təqdim edir. Selenium test edənlərə və tərtibatçılara veb brauzerlərlə qarşılıqlı əlaqədə olmaq və veb proqramların davranışını yoxlamaq üçün müxtəlif proqramlaşdırma dillərində skriptlər yazmağa imkan verir. [21]

1. Selenium VebDriver:

Selenium VebDriver Seleniumun əsas komponentidir. Veb brauzerləri ilə qarşılıqlı əlaqə üçün proqramlaşdırma interfeysi təmin edir. VebDriver brauzerə məxsus sürücüdən (məsələn, ChromeDriver, GeckoDriver və s.) istifadə edərək birbaşa brauzerlə əlaqə saxlayır. VebDriver test skriptləriniz və brauzer arasında körpü rolunu oynayır. [10]

- Test Skriptinin Yaradılması: Java, Python, C#, Ruby və ya JavaScript kimi proqramlaşdırma dillərindən istifadə edərək testçilər veb elementləri ilə qarşılıqlı

əlaqədə olmaq və veb proqramda hərəkətlər etmək üçün VebDriver əmrlərindən istifadə edən test skriptləri yazır.

- Brauzer Əlaqəsi: VebDriver VebDriver API istifadə edərək brauzer sürücüsü ilə əlaqə qurur. O, brauzer sürücüsünə əmrlər göndərir, sonra isə onları brauzerdəki hərəkətlərə çevirir. Məsələn, VebDriver brauzerə URL-i açmağı, elementə klik etməyi, forma sahələrini doldurmağı və ya ekran görüntülərini çəkməyi əmr edə bilər.

- Brauzer üçün Xüsusi Sürücülər: Selenium VebDriver əlaqə yaratmaq və brauzeri idarə etmək üçün hər bir brauzer üçün xüsusi sürücü tələb edir (məsələn, Google Chrome üçün ChromeDriver, Mozilla Firefox üçün GeckoDriver). Bu drayverlər VebDriver və brauzer arasında vasitəçi kimi çıxış edərək, brauzerə xas funksionallığı təmin edir.

- Çarpaz Brauzer Testi: Selenium VebDriver müxtəlif brauzerlər arasında ardıcıl proqramlaşdırma interfeysi təmin etməklə, cross-brauzer testinə imkan verir. VebDriver ilə yazılmış test skriptləri müxtəlif brauzerlərdə icra oluna bilər ki, bu da sizə veb proqramınızın davranışını və uyğunluğunu yoxlamağa imkan verir.

Ümumiyyətlə, Selenium veb brauzerlərin avtomatlaşdırılması üçün güclü və çevik çərçivə təmin edir. O, müxtəlif sınaq ehtiyaclarını və ssenarilərini təmin etmək üçün VebDriver, IDE və Grid kimi müxtəlif komponentlər təklif edir, test edənlərə və tərtibatçılara veb tətbiqləri üçün möhkəm və səmərəli avtomatlaşdırılmış testlər yaratmağa imkan verir.

2.3. Veb tətbiqi testləri üçün Selenium tətbiqi

Selenium veb-brauzerləri avtomatlaşdırmaq üçün istifadə edilən məşhur açıq mənbəli çərçivədir. O, müxtəlif brauzerlər və platformalarda veb proqramları sınamaq üçün bir sıra alətlər və kitabxanalar təqdim edir. Selenium Java, Python, C#, Ruby və s. kimi müxtəlif proqramlaşdırma dillərində test skriptləri yazmağa imkan verir.

Veb tətbiqi testi üçün Seleniumun necə tətbiq ediləcəyinə dair addım-addım təlimat budur:

1. Selenium quraşdırın: Veb brauzerlərlə qarşılıqlı əlaqə üçün əsas komponent olan Selenium VebDriver-ı quraşdırmaqla başlayın. Siz onu pip (Python üçün), NuGet

(C# üçün) kimi paket menecerlərindən istifadə edərək və ya birbaşa VebDriver icra edilə bilən fayllarını endirməklə quraşdıra bilərsiniz.

2. Proqramlaşdırma dilini seçin: Selenium bir çox proqramlaşdırma dillərini dəstəkləyir. Ən rahat olduğunuzu və ya layihə tələblərinizə uyğun olanı seçin.

3. İnkişaf mühitini qurun: Selenium skriptlərini yazmaq və icra etmək üçün üstünlük verdiyiniz inteqrasiya olunmuş inkişaf mühitini (IDE) qurun. Ümumi seçimlərə Eclipse, IntelliJ IDEA, Visual Studio və ya PyCharm daxildir.

4. Yeni layihə yaradın: Seçdiyiniz IDE-də yeni layihə yaradın və onu Selenium VebDriver-dən istifadə etmək üçün konfiqurasiya edin. İstifadə etdiyiniz proqramlaşdırma dili əsasında lazımi asılılıqları və ya paketləri əlavə edin.

5. VebDriver-ı konfiqurasiya edin: Seçdiyiniz brauzerlə işləmək üçün VebDriver-i qurun. VebDriver ChromeDriver, GeckoDriver (Firefox üçün), SafariDriver və s. kimi müxtəlif brauzerə xas tətbiqləri təmin edir. Brauzeriniz üçün müvafiq icra olunan sürücünü yükləyin və onu layihənizdə konfiqurasiya edin.

6. Test skriptlərini yazın: Selenium VebDriver API istifadə edərək test skriptlərini yazmağa başlayın. Bu API veb elementləri ilə qarşılıqlı əlaqə yaratmaq, istifadəçi hərəkətlərini simulyasiya etmək və veb səhifələrdən məlumat əldə etmək üçün metodlar və siniflər təqdim edir. Siz bu üsullardan URL-lərə getmək, formalarla əlaqə saxlamaq, düymələri klikləmək, mətn sahələrini doldurmaq, səhifə məzmununu yoxlamaq və digər hərəkətləri yerinə yetirmək üçün istifadə edə bilərsiniz.

7. Test skriptlərini təkmilləşdirin: Siz müxtəlif Selenium xüsusiyyətlərindən istifadə etməklə test skriptlərinizi təkmilləşdirə bilərsiniz, məsələn:

- Elementlərin yerləşdirilməsi: Selenium veb-səhifədə elementləri tapmaq üçün müxtəlif strategiyalar təqdim edir, məsələn ID, sinif adı, CSS seçicisi, XPath və s.

- Fərqli idarəetmə vasitələrinin idarə edilməsi: Siz VebDriver metodlarından istifadə edərək onay qutuları, radio düymələri, açılan siyahılar, xəbərdarlıqlar, pop-uplar və digər UI elementləri ilə əlaqə saxlaya bilərsiniz.

- Gözləmələr: Səhifənin yüklənməsi gecikmələrini və ya skriptin icrası ilə səhifənin göstərilməsi arasında sinxronizasiya problemlərini həll etmək üçün açıq və ya gizli gözləmələri həyata keçirin.

- Test təsdiqləri: Gözlənilən nəticələri təsdiqləmək və faktiki nəticələri gözlənilənlərlə müqayisə etmək üçün proqramlaşdırma dilinizin sınaq çərçivəsindən (məsələn, JUnit, NUnit, TestNG) təsdiqlərdən istifadə edin.

- Test məlumatlarının idarə edilməsi: Məlumata əsaslanan testi həyata keçirmək üçün xarici fayllardan (məsələn, Excel, CSV) və ya verilənlər bazalarından oxumaqla test məlumatlarını idarə edin.

- Test dəstləri və hesabatlar: Test skriptlərinizi test paketləri şəklində təşkil edin və testin icrası nəticələrini və uğursuzluqları izləmək üçün hesabatlar yaradın.

8. Test skriptlərini yerinə yetirin: Test skriptlərinizi fərdi olaraq və ya test paketinin bir hissəsi kimi işlədin. İcra prosesini izləyin və sınaq zamanı rast gəlinən hər hansı uğursuzluq və ya səhvləri müşahidə edin.

9. Test nəticələrini təhlil edin: Veb tətbiqinizdə hər hansı problem və ya qüsuru müəyyən etmək üçün test nəticələrini təhlil edin. Selenium əlavə araşdırma üçün ekran görüntülərini, qeydləri və digər sazlama məlumatlarını çəkmək üçün API təmin edir.

10. Test skriptlərini qoruyun və yeniləyin: Veb proqramınız inkişaf etdikcə, tətbiqin funksionallığında və ya istifadəçi interfeysində dəyişikliklərə uyğunlaşmaq üçün test skriptlərinizi yeniləyin.

Unutmayın ki, Selenium Veb tətbiqi testləri üçün güclü bir vasitədir, lakin bu, ümumi sınaq prosesinin yalnız bir hissəsidir. Siz həmçinin Selenium-u TestNG və ya JUnit kimi digər test çərçivələri ilə inteqrasiya etməyi və davamlı inteqrasiya və test avtomatlaşdırılması üçün Jenkins kimi əlavə vasitələrdən istifadə etməyi düşünə bilərsiniz.

2.4. Test edilən veb proqrama ümumi baxış

Test edilən veb tətbiqi ilə bağlı hərtərəfli icmalı təmin etmək üçün aşağıdakı aspektləri nəzərə almaq vacibdir:

1. Məqsəd və funksionallıq: Veb proqramın məqsədini və onun əsas funksionallığını təsvir edin. Bu, e-ticarət platforması, sosial media saytı, məzmun idarəetmə sistemi və ya başqa bir şeydir? İstifadəçilərə təklif etdiyi əsas xüsusiyyətləri və funksiyaları təsvir edin.

2. Hədəf auditoriyası: Veb tətbiqi üçün hədəf auditoriyanı müəyyən edin. O, ümumi istifadəçilər, xüsusi sənayelər və ya niş bazar üçün nəzərdə tutulub? Hədəf auditoriyasını başa düşmək test yanaşmasını və prioritetləri formalaşdırmağa kömək edir.

3. Texnologiya yığını: Veb tətbiqini inkişaf etdirmək üçün istifadə olunan texnologiyaları sadalayın. Buraya proqramlaşdırma dilləri (məsələn, Java, Python, PHP), çərçivələr (məsələn, Ruby on Rails, Django, Laravel), verilənlər bazası (məsələn, MySQL, PostgreSQL) və hər hansı digər müvafiq alətlər və ya kitabxanalar daxil ola bilər.

4. Dəstəklənən brauzerlər və platformalar: Veb tətbiqinin dəstəklədiyi brauzerləri və platformaları müəyyənləşdirin. Buraya Chrome, Firefox, Safari və Internet Explorer kimi məşhur brauzerlər, həmçinin Windows, macOS, Linux və mobil platformalar (Android, iOS) kimi əməliyyat sistemləri daxildir.

5. İstifadəçi interfeysi (UI): Veb tətbiqinin istifadəçi interfeysi dizaynını qiymətləndirin. O, həssas, intuitiv və vizual olaraq cəlbedicidirmi? Dizaynı, naviqasiyanı, formaları, düymələri, menyuları və digər UI elementlərini nəzərdən keçirin. UI-nin müəyyən edilmiş dizayn prinsiplərinə və təlimatlarına uyğun olub olmadığını qiymətləndirin.

6. Performans tələbləri: Veb tətbiqi üçün performans gözləntilərini müəyyən edin. Səhifənin yüklənmə vaxtları, istifadəçi hərəkətləri üçün cavab müddətləri, eyni vaxtda istifadəçilərin idarə edilməsi və ümumi sistemin genişlənməsi kimi amilləri nəzərdən keçirin. Bu aspektlər tətbiqin istənilən performans standartlarına cavab verməsini təmin etmək üçün çox vacibdir.

7. Təhlükəsizlik mülahizələri: Veb tətbiqinin təhlükəsizlik tələblərini və xüsusiyyətlərini nəzərdən keçirin. Onun autentifikasiya, avtorizasiya, şifrələmə və ümumi veb zəifliklərindən qorunma kimi tədbirləri həyata keçirib-etdirmədiyini qiymətləndirin (məsələn, saytlar arası skript, SQL inyeksiyası, CSRF). Tətbiqin riayət etməli olduğu hər hansı xüsusi təhlükəsizlik standartlarını və ya qaydaları anlayın.

8. İnteqrasiya nöqtələri: Veb tətbiqinin inteqrasiya etdiyi hər hansı xarici sistemləri, xidmətləri və ya API-ləri müəyyən edin. Buraya ödəniş şlüzləri, üçüncü

tərəf API-ləri, verilənlər bazası və ya digər arxa sistemlər daxil ola bilər. Testi asanlaşdırmaq üçün inteqrasiya nöqtələrinin yaxşı sənədləşdirilmiş və başa düşüldüyünə əmin olun.

9. Test məlumatlarına dair tələblər: Veb proqramında istifadə olunan məlumat növlərini və test məlumatları üçün hər hansı xüsusi tələbləri müəyyən edin. Buraya nümunə istifadəçi hesabları, məhsul məlumatları və ya müxtəlif istifadə nümunələrini əhatə etmək üçün simulyasiya edilmiş ssenarilər daxil ola bilər. Verilənlərin məxfiliyinə ehtiyacı nəzərə alın və sınaq zamanı həssas məlumatların müvafiq şəkildə idarə olunmasını təmin edin.

10. Məlum məsələlər və ya məhdudiyyətlər: Veb proqramında hər hansı məlum məsələlər, məhdudiyyətlər və ya narahatlıq sahələri haqqında məlumat toplayın. Buraya keçmiş səhv hesabatları, müştəri rəyi və ya məlum performans darboğazları daxil ola bilər. Bu aspektləri başa düşmək sınaq səylərini cəmləşdirməyə və əvvəlki problemlərin həll edilib-edilmədiyini təsdiq etməyə kömək edir.

Bu aspektləri nəzərə alaraq, siz effektiv test strategiyaları tərtib etməyə, test işlərinə üstünlük verməyə və hərtərəfli test əhatəsini təmin etməyə imkan verən sınaqdan keçirilən veb tətbiqi haqqında hərtərəfli anlayış yarada bilərsiniz.

2.5. Test ssenarilərinin və test işlərinin seçilməsi

Selenium ilə inteqrasiya üçün test ssenariləri və test nümunələri seçərkən aşağıdakı amilləri nəzərə almaq vacibdir:

1. Tələblərin əhatə dairəsi: Layihə tələblərini nəzərdən keçirin və veb tətbiqinin əsas funksiyalarını və xüsusiyyətlərini müəyyənləşdirin. Hər bir tələbi müvafiq test ssenarilərinə və test vəziyyətlərinə uyğunlaşdırın. Bu, testlərinizin tətbiqin gözlənilən davranışını əhatə etməsini təmin edir.

2. Biznes prioriteti: Biznesə təsir və kritikliyə əsaslanaraq sınaq ssenarilərinə və sınaq işlərinə üstünlük verin. Tətbiqin işləməsi üçün vacib olan və ya istifadəçi təcrübəsinə, gəlir əldə etməyə və ya tənzimləyicilərə uyğunluğa əhəmiyyətli təsir göstərən yüksək prioritet sahələrə diqqət yetirin.

3. Risk əsaslı test: Veb tətbiqinin müxtəlif hissələri ilə bağlı riskləri qiymətləndirin. Mürəkkəb funksionallıqlar, təhlükəsizlik baxımından kritik funksiyalar və ya problem tarixçəsi olan sahələr kimi yüksək riskli sahələrə müraciət edən sınaq ssenariləri və sınaq işlərinə üstünlük verin.

4. Başdan-başa ssenarilər: Tətbiq vasitəsilə bir-birinin ardınca istifadəçi iş axınlarını və ya istifadəçi səyahətlərini əhatə edən sınaq ssenarilərini daxil edin. Bu ssenarilər real dünyada istifadəni simulyasiya edir və naviqasiya, məlumat axını və inteqrasiya nöqtələri ilə bağlı problemləri aşkar etməyə kömək edir.

5. Müsbət və mənfi test: Həm müsbət, həm də mənfi ssenariləri nəzərdən keçirin. Müsbət testlər tətbiqin normal şəraitdə gözlənildiyi kimi davrandığını, mənfi testlər isə müvafiq səhvlərin idarə edilməsi və istisnaların idarə edilməsi mexanizmlərinin mövcud olduğunu təsdiqləyir.

6. Sərhəd və kənar hallar: Maksimum və minimum qiymətlər, boş və ya sıfır girişlər və müəyyən edilmiş uzunluqları aşan girişlər kimi giriş dəyərlərinin sərhədlərini və limitlərini sınaqdan keçirən sınaq hallarını daxil edin. Bu, məlumatların yoxlanılması, daşqın və ya sərhəd şərtləri ilə bağlı problemləri aşkar etməyə kömək edir.

7. Məlumata əsaslanan test: Mümkünsə, eyni test ssenarisini yerinə yetirmək üçün müxtəlif test məlumat dəstlərindən istifadə etməklə dataya əsaslanan test üsullarını daxil edin. Bu, müxtəlif məlumat daxiletmələri ilə tətbiqin davranışını təsdiqləməyə kömək edir və hərtərəfli əhatəni təmin edir.

8. Brauzerlər arası və platformalararası sınaq: Veb tətbiqi tərəfindən dəstəklənən müxtəlif brauzerləri və platformaları əhatə edən test nümunələrini daxil edin. Selenium müxtəlif mühitlərdə uyğunluq və ardıcıllığı təmin etməyə imkan verən çoxsaylı brauzerlər və platformalarda testlər aparmaq imkanı verir.

9. Səhvlərin idarə edilməsi və bərpası: Səhvlərin idarə edilməsi və bərpa mexanizmlərinə diqqət yetirən sınaq ssenarilərini daxil edin. Bu kateqoriyadakı test halları tətbiqin istisnaları zərif şəkildə idarə etdiyini, müvafiq səhv mesajlarını göstərdiyini və məlumat itkisi və ya sistem qeyri-sabitliyi olmadan bərpa olunduğunu yoxlamağa kömək edir.

10. İstifadəyə yararlılıq və əlçatanlıq testi: Veb tətbiqinin yararlılığını və əlçatanlığını qiymətləndirən test ssenarilərini və sınaq nümunələrini daxil edin. Əlçatanlıq qaydalarına (məsələn, WCAG) uyğunluğu yoxlayın və tətbiqin intuitiv, istifadəçi dostu olduğunu və müsbət istifadəçi təcrübəsi təmin etdiyini təsdiqləyin.

Müvafiq test ssenarilərini və sınaq nümunələrini müəyyən etdikdən sonra Selenium VebDriver API-dən istifadə edərək lazımi test skriptlərini tətbiq etməklə onları Selenium ilə inteqrasiya edə bilərsiniz. Veb elementləri ilə qarşılıqlı əlaqə yaratmaq, istifadəçi hərəkətlərini simulyasiya etmək və yoxlamaları yerinə yetirmək üçün müvafiq VebDriver metodlarından istifadə edin. [22]

TestNG, JUnit və ya PyTest kimi alətlərlə Seleniumun test çərçivəsi inteqrasiyasından istifadə edərək, test ssenarilərinizi və test işlərinizi məntiqi qruplara və ya test paketlərinə təşkil edin. Bu çərçivələr testin idarə edilməsi, paralel icra, hesabat və test konfiqurasiyası üçün əlavə imkanlar təmin edir.

Test işlərinizin modul, saxlanıla bilən və təkrar istifadə edilə bilən olduğundan əmin olun. Test məntiqini səhifə strukturundan ayırmaq üçün Səhifə Obyekt Modeli (POM) dizayn nümunəsindən istifadə edərək test skriptlərinizi daha möhkəm və davamlı hala gətirir.

Test ssenarilərinizi və test işlərinizi Selenium ilə birləşdirərək, testləri səmərəli şəkildə yerinə yetirmək, veb tətbiqinin davranışını yoxlamaq və hər hansı problemi və ya reqressiyaları vaxtında müəyyən etmək üçün onun avtomatlaşdırma imkanlarından istifadə etmək olar.

2.6. Selenium skriptlərinin yazılması və icrası

Selenium skriptlərinin yazılması və icrası aşağıdakı addımları əhatə edir:

1. İnkişaf mühiti qurulması:

- Tələb olunan proqramlaşdırma dilini (məsələn, Java, Python) və müvafiq IDE-ni (məsələn, Eclipse, PyCharm) quraşdırmaq lazımdır.

- Layihəyə lazımi asılılıqlar və ya paketlər əlavə etməklə Selenium VebDriver-I quraşdırmaq lazımdır.

2. VebDriver-in konfiqurasiya edilməsi:

- Avtomatlaşdırılmalı olan brauzer üçün müvafiq icra olunan VebDriver proqramın yüklənməlidir (məsələn, ChromeDriver, Firefox üçün GeckoDriver).

- Layihəndə yerinə yetirilən VebDriver proqramını onun yerini göstərərək və ya sistemin PATH dəyişəninə əlavə etməklə konfiqurasiya edmək lazımdır.

3. Yeni test skripti yaradılması:

- IDE-də yeni test skript faylı yaratmaq və lazımi Selenium VebDriver siniflərini və ya modullarını idxal etmək lazımdır.

- İstənilən tələb olunan dəyişənləri və ya sabitləri, məsələn, veb tətbiqinin URL-i və ya veb elementləri üçün lokatorlar qurmaq lazımdır.

4. Test kodunu yazılmalı:

- Veb elementləri ilə qarşılıqlı əlaqədə olmaq və veb proqramda hərəkətlər etmək üçün VebDriver metodlarından istifadə etmək lazımdır. Məsələn, xüsusi URL-ə getmək üçün `driver.get("URL"), veb elementləri tapmaq üçün `driver.findElement(By.locator("yerləşdirici"))` və `webElement.sendKeys("mətn") istifadə edin.) daxiletmə sahələrinə mətn daxil edilməlidir.

- Düymələrə klikləmək, açılan seçimləri seçmək, formaları təqdim etmək və ya sürüşdürmək kimi istifadəçi hərəkətlərini simulyasiya etmək üçün müxtəlif VebDriver metodlarından istifadə etmək lazımdır.

- Veb tətbiqinin gözlənilən davranışını təsdiqləmək üçün təsdiq və ya yoxlama nöqtələrini həyata keçirmək lazımdır. Məsələn, faktiki və gözlənilən dəyərləri müqayisə etmək üçün seçilən proqramlaşdırma dilinin sınaq çərçivəsindən (məsələn, Java-da `assertEquals(expected, actual)`) və ya Selenium-un `webElement.getText()` kimi daxili təsdiqləmə metodlarından istifadə etmək lazımdır.

5. Test skriptlərini təkmilləşdirilməsi:

- Skript icrası və səhifənin yüklənməsi arasında sinxronizasiya məsələlərini həll etmək üçün açıq və ya gizli gözləmələri həyata keçirilməlidir. Məsələn, skriptlə davam etməzdən əvvəl xüsusi şərtləri gözləmək üçün `WebDriverWait` və `ExpectedConditions` sinfindən istifadə etmək lazımdır.

- VebDriver-in seçim siniflərindən istifadə edərək brauzerə xas imkanlardan və ya parametrlərdən istifadə edilir. Məsələn, siz brauzer seçimləri təyin edilə, kukiləri idarə edilə və ya pop-up pəncərələri idarə edilə bilər.

- Sazlama və səhv təhlilini təkmilləşdirmək üçün ekran görüntülərinin çəkilməsi və ya giriş kimi qabaqcıl üsullardan istifadə edilməlidir.

6. Test skriptini yerinə yetirilməlidir:

- Skript faylını icra etməklə və ya IDE-nin test proqramı funksiyasından istifadə etməklə IDE-dən test skriptini işə salınmalıdır.

- İcra prosesini izlənməli və sınaq zamanı rast gəlinən hər hansı nasazlıqları və ya səhvləri müşahidə edilməlidir.

7. Test nəticələrini təhlil edilməsi:

- Selenium tərəfindən bildirilən hər hansı uğursuzluq və ya səhvləri müəyyən etmək üçün testin icrası nəticələrini nəzərdən keçirmıək lazımdır.

- Ekran görüntüləri, xəta mesajları və ya sazlama təfərrüatları kimi sınaq əməliyyatı haqqında əlavə məlumatları əldə etmək üçün qeyd və ya hesabat mexanizmlərindən istifadə etmək lazımdır.

- Veb tətbiqindəki problemləri və ya reqressiyaları müəyyən etmək və müvafiq tədbirlər görmək üçün nəticələri təhlil etmək lazımdır.

8. Test skriptlərini qorumaq və yeniləmək lazımdır:

- Veb tətbiqi inkişaf etdikcə, tətbiqin funksionallığında və ya istifadəçi interfeysində dəyişikliklərə uyğunlaşmaq üçün test skriptlərini yenilənməlidir.

- Yeni test ssenarilərini daxil etmək, əhatə dairəsini yaxşılaşdırmaq və skript performansını optimallaşdırmaq üçün test skriptlərini mütəmadi olaraq nəzərdən keçirmək və təkmilləşdirmək lazımdır.

Selenium skriptlərinizdə istisnaları və səhvləri müvafiq qaydada idarə etməyi və möhkəmliyi və zərif xətaların bərpasını təmin etmək üçün müvafiq istisnaların idarə edilməsi üsullarından istifadə etmək lazımdır.

Bu addımları yerinə yetirməklə veb proqramların sınaqdan keçirilməsini avtomatlaşdırmaq üçün Selenium skriptləri effektiv şəkildə yazıla və icra edılə bilər.

2.7. Selenium əsaslı avtomatlaşdırılmış sınaqların qiymətləndirilməsi

Selenium əsaslı avtomatlaşdırılmış sınaq qiymətləndirmə üçün bir sıra üstünlüklər və mülahizələr təklif edir:

Üstünlüklərini qeyd emək lazımdır:

1. Test əhatə dairəsi: Selenium veb elementləri ilə qarşılıqlı əlaqəni avtomatlaşdırmaq, funksionallığı yoxlamaq və müxtəlif brauzerlər və platformalarda gözlənilən davranışları təsdiqləməklə geniş test əhatəsinə imkan verir. Bu, veb tətbiqinin hərtərəfli sınaqdan keçirilməsini təmin edir.

2. Vaxta və xərclərə qənaət: Selenium ilə avtomatlaşdırılmış sınaq əl ilə sınaqla müqayisədə təkrar test işlərinin icrası üçün tələb olunan vaxtı və səyi azaldır. Bu, testin daha sürətli icrasına, səhvlərin erkən aşkarlanmasına və daha sürətli əks əlaqə dövrələrinə imkan verir ki, bu da uzun müddətdə xərclərə qənaət edir.

3. Təkmilləşdirilmiş dəqiqlik: Avtomatlaşdırılmış testlər əl ilə sınaq zamanı baş verə biləcək insan səhvlərini və uyğunsuzluqları aradan qaldırır. Selenium veb elementləri ilə qarşılıqlı əlaqə üzərində dəqiq nəzarəti təmin edir və nəticədə daha dəqiq və etibarlı test nəticələri əldə edilir.

4. Reqressiya testi: Selenium reqressiya testi üçün çox uyğundur, burada dəyişikliklərin heç bir gözlənilməz yan təsir və ya reqressiyaya səbəb olmadığından əmin olmaq üçün dəyişikliklər və ya yeniləmələrdən sonra mövcud funksiyalar yenidən sınaqdan keçirilir.

5. Çarpaz brauzer və platformalararası uyğunluq: Seleniumun müxtəlif brauzerlərdə (Chrome, Firefox, Safari və s.) və platformalarda (Windows, macOS, Linux və s.) veb proqramlarını sınaqdan keçirmək qabiliyyəti müxtəlif proqramlarda uyğunluq və ardıcıl davranışı təmin etməyə kömək edir. istifadəçi mühitləri.

6. Test çərçivələri ilə inteqrasiya: Selenium TestNG, JUnit və ya PyTest kimi məşhur sınaq çərçivələri ilə inteqrasiya oluna bilər ki, bu da təkmilləşdirilmiş test idarəçiliyinə, paralel icraya, hesabat verməyə və davamlı inteqrasiya (CI) boru kəmərləri ilə inteqrasiyaya imkan verir.

Çatışmazlıqları qeyd etmək lazımdır:

1. İlkin quraşdırma və öyrənmə əyrisi: Selenium-un qurulması və avtomatlaşdırılmış testə başlaması bəzi ilkin səylər və öyrənmə əyrisi tələb edə bilər, xüsusən də Selenium ilə istifadə olunan çərçivə və ya proqramlaşdırma dillərində yeni olanlar üçün.

2. Texniki xidmət xərcləri: Veb proqramdakı dəyişikliklərə uyğunlaşmaq üçün avtomatlaşdırılmış testlər saxlanılmalı və yenilənməlidir. Tətbiq inkişaf etdikcə test skriptləri yeni xüsusiyyətlərə, UI dəyişikliklərinə və ya tətbiqdəki təkmilləşdirmələrə uyğunlaşmaq üçün modifikasiya tələb edə bilər.

3. Test sabitliyi və etibarlılığı: Selenium testləri veb tətbiqinin UI və ya əsas HTML strukturunda dəyişikliklərdən təsirlənə bilər. Kiçik UI dəyişiklikləri və ya dinamik elementlər sabitliyi və etibarlılığı qorumaq üçün test skriptinə düzəlişlər və yeniləmələr tələb edən test uğursuzluqlarına səbəb ola bilər.

4. Qeyri-veb texnologiyaları üçün məhdud dəstək: Selenium ilk növbədə veb proqramların sınaqdan keçirilməsinə yönəlib və masaüstü proqramlar və ya mobil proqramlar kimi digər texnologiyaların sınaqdan keçirilməsi üçün geniş dəstək təklif etmir.

5. Testin icra müddəti: Test işlərinin mürəkkəbliyindən və sayından asılı olaraq, Selenium əsaslı avtomatlaşdırılmış testlər üçün icra müddəti əllə sınaqdan daha uzun ola bilər. Ümumi test müddətini minimuma endirmək üçün test skriptlərinin optimallaşdırılmasına və icrasına diqqət yetirilməlidir.

6. Brauzerlərdən və VebDriver-dən asılılıq: Selenium testləri veb brauzerlərlə qarşılıqlı əlaqə üçün VebDriver və brauzerə xas drayverlərə əsaslanır. Brauzerlərə və ya VebDriver versiyalarına yeniləmələr test skriptlərinə və sürücü uyğunluğuna müvafiq yeniləmələr tələb edə bilər.

7. Performans testi məhdudiyyətləri: Selenium səhifənin yüklənmə vaxtlarını ölçə və veb elementləri ilə əlaqə qura bilsə də, güclü performans testi imkanları təmin etməyə bilər. Dərin performans testi üçün əlavə alətlər və üsullar tələb oluna bilər.

Selenium əsaslı avtomatlaşdırılmış sınaqların qiymətləndirilməsi layihənin xüsusi ehtiyaclarını, tələblərini və kontekstini nəzərə almalıdır. Seleniumun layihənin

avtomatlaşdırılmış sınaq strategiyası üçün düzgün seçim olub-olmadığını müəyyən etmək üçün faydaları və mülahizələri ölçmək vacibdir.

2.8. Avtomatlaşdırılmış testin əl testi ilə müqayisəsi

Avtomatlaşdırılmış sınaq və əllə sınaq proqram təminatının sınaqdan keçirilməsinə iki fərqli yanaşmadır, hər birinin öz güclü və zəif tərəfləri var. Budur ikisi arasında bir müqayisə:

Avtomatlaşdırılmış sınaq: Səmərəlilik və sürət, Təkrarlanan və reqressiya testi, Test əhatə dairəsi, Dəqiqlik və etibarlılıq, Paralel icra və miqyaslılıq.

Manual Test: Kəşfiyyat və xüsusi sınaq, İstifadəyə yararlılıq və istifadəçi təcrübəsinin qiymətləndirilməsi, İstifadəçi qavrayışının yoxlanılması, İstifadəçi interfeysi (UI) dizaynının yoxlanılması, Dəyişikliklərə uyğunlaşma, Erkən sınaq və istifadəyə yararlılıq rəyi.

Qeyd etmək vacibdir ki, avtomatlaşdırılmış sınaq və əl testi bir-birini istisna etmir. Onlar layihənin xüsusi ehtiyaclarına və kontekstinə əsasən birləşdirilə bilər. Avtomatlaşdırılmış sınaq effektivlik, təkrarlanma qabiliyyəti və geniş əhatə dairəsi baxımından üstün olsa da, əllə sınaq test prosesinə insan anlayışını, uyğunlaşma qabiliyyətini və subyektiv qiymətləndirməsini gətirir.

2.9. Avtomatlaşdırılmış sınaqların nəticələrinin təhlili

Avtomatlaşdırılmış testin nəticələrinin təhlili proqram təminatının sınaqdan keçirilməsi prosesində mühüm addımdır. Bu, testin icra nəticəsini nəzərdən keçirmək və şərh etmək və hər hansı problem və ya anomaliyaları müəyyən etməkdən ibarətdir. Avtomatlaşdırılmış testin nəticələri üçün təhlil çərçivəsi budur:

1. Testin İcrası Xülasəsi:

- İcra edilmiş, keçilmiş və uğursuz sınaqların sayı daxil olmaqla, ümumi sınaq icrası xülasəsini nəzərdən keçirin. Bu, testin əhatə dairəsi və müvəffəqiyyət dərəcəsi haqqında ilkin anlayış verir.

2. Uğursuzluğun təhlili:

- Uğursuz sınaqları müəyyən edin və uğursuzluğun səbəblərini araşdırın. Test çərçivəsi və ya aləti tərəfindən bildirilən səhv mesajlarını, yığın izlərini və ya təsdiqləmə uğursuzluqlarını axtarın.

3. Qüsurların İdarə Edilməsi:

- Müəyyən edilmiş nasazlıqlar üçün qüsur hesabatları yaradın və ya yeniləyin. Uğursuzluq haqqında ətraflı məlumat, təkrar istehsal addımları, gözlənilən və faktiki nəticələr və hər hansı dəstəkləyici sübut (məsələn, ekran görüntüləri, qeydlər) daxil edin.

4. Səhv diaqnozu:

- Uğursuzluqların əsas səbəbini anlamaq üçün xəta mesajlarını və yığın izlərini təhlil edin. Uğursuzluqların funksional qüsurlar, ətraf mühit problemləri, məlumat uyğunsuzluqları və ya sınaq skriptindəki nasazlıqlardan qaynaqlandığını müəyyənləşdirin.

5. Bug Triage:

- Bildirilən uğursuzluqları yoxlamaq üçün tərtibatçılar, məhsul sahibləri və digər maraqlı tərəflərlə əməkdaşlıq edin. Təsiri, ciddiliyi və biznes prioritetləri əsasında qüsurları prioritetləşdirin.

6. Reqressiya təhlili:

- Son dəyişikliklər və ya yeni funksiyalar tərəfindən təqdim edilən hər hansı reqressiya qüsurlarını müəyyən edin. Tətbiqin davranışının gözlənilmədən dəyişdiyi sahələri müəyyən etmək üçün cari test nəticələrini əvvəlki sınaq sınaqları ilə müqayisə edin.

7. Test Metrikləri və Hesabatları:

- Ümumi testin icrası, testin əhatə dairəsi və qüsur meylləri haqqında məlumat vermək üçün test ölçüləri və hesabatları yaradın.

8. Əlaqə və Davamlı Təkmilləşdirmə:

- Test nəticələri və qüsurlar haqqında kod keyfiyyətinin, testin əhatə dairəsinin və ya test skriptlərinin təkmilləşdirilməsi üçün sahələri vurğulayaraq inkişaf qrupuna rəy bildirin.

Avtomatlaşdırılmış sınaqların nəticələrinin müntəzəm təhlili sınaq prosesinin effektivliyini və etibarlılığını təmin edir. O, qüsurları erkən müəyyən etməyə kömək edir, tətbiqi təkmilləşdirmək üçün dəyərli rəy verir və yüksək keyfiyyətli proqram təminatının çatdırılmasını təmin edir.

2.10. Müxtəlif Selenium tətbiqlərinin müqayisəsi

Fərqli xüsusiyyətlər, proqramlaşdırma dili dəstəyi və veb avtomatlaşdırılmasına yanaşmalar təmin edən çoxlu Selenium tətbiqləri mövcuddur. Budur üç məşhur Selenium tətbiqinin müqayisəsi:

1. Selenium VebDriver:

- Dil dəstəyi: Selenium VebDriver Java, Python, C#, Ruby və JavaScript daxil olmaqla bir çox proqramlaşdırma dillərini dəstəkləyir və tərtibatçılara üstünlük verdiyi dilləri seçməyə imkan verir.

- Çarpaz Brauzer Uyğunluğu: VebDriver ChromeDriver, GeckoDriver və Microsoft Edge üçün VebDriver kimi brauzerə xas drayverlər vasitəsilə veb brauzerlərlə birbaşa əlaqəni təmin edir. Bu, müxtəlif brauzer versiyaları və platformaları üzrə cross-brauzer testinə imkan verir.

- Çeviklik: VebDriver brauzerin qarşılıqlı əlaqəsi və veb elementləri üzərində birbaşa nəzarəti təmin edərək veb avtomatlaşdırılmasına daha çevik və aşağı səviyyəli yanaşma təklif edir.

- W3C Standardı: Selenium VebDriver, müxtəlif Selenium tətbiqləri arasında uyğunluq və ardıcıllığı təmin edən W3C VebDriver spesifikasiyasına əməl edir. [23]

2. Selenium IDE:

- Qeyd və Oynatma: Selenium IDE ilk növbədə istifadəçilərə brauzerlə qarşılıqlı əlaqəni qeyd edərək test skriptləri yaratmağa imkan verən qeyd və oxutma vasitəsidir. O, HTML, Java, C# və ya Python-da test skriptləri yaradır.

- Sürətli Test Yaradılması: Selenium IDE, avtomatlaşdırılmış testləri tez bir zamanda yaratmaq üçün sadə və intuitiv interfeys təqdim edərək onu minimal proqramlaşdırma biliyi və ya təcrübəsi olan istifadəçilər üçün uyğun edir.

- Məhdud Dil Dəstəyi: Selenium IDE test skriptinin yaradılması üçün məhdud proqramlaşdırma dilləri dəstini dəstəkləyir, bu da müxtəlif dillərlə işləyən komandalar üçün istifadəsini məhdudlaşdıra bilər.

- Məhdud Çeviklik: Selenium IDE VebDriver ilə müqayisədə məhdud çevikliyə malikdir, çünki o, ilk növbədə qeyd və oxutma funksionallığına diqqət yetirir. O, mürəkkəb sınaq ssenariləri üçün tələb olunan qabaqcıl xüsusiyyətləri və ya imkanları təmin etməyə bilər.

3. Selenium Grid:

- Paylanmış Test: Selenium Grid, testlərin eyni vaxtda birdən çox maşın və ya mühitdə yerinə yetirilməsinə imkan verməklə paylanmış testlərə imkan verir. Bu, testin daha sürətli icrasına və genişlənməyə nail olmağa kömək edir.

- Paralel Test İcrası: Selenium Grid paralel test icrasını dəstəkləyir, müxtəlif maşınlarda, brauzerlərdə və ya platformalarda eyni vaxtda birdən çox testin aparılmasına imkan verir.

- Cross-Browser və Cross-Platform Testing: Selenium Grid müxtəlif brauzerlər və platformalar arasında testləri yaymaq imkanı təmin etməklə, brauzerlər arası və çarpaz platforma testini asanlaşdırır.

- Kompleks Quraşdırma: Selenium Grid-in qurulması və konfiqurasiyası VebDriver və ya Selenium IDE ilə müqayisədə əlavə səy tələb edir, çünki bu, mərkəzin konfiqurasiyasını və testin icrası üçün çoxlu qovşaqların qeydiyyatını nəzərdə tutur.

- Baxım: Selenium Grid şəbəkə infrastrukturunun mövcudluğunu və sabitliyini təmin etmək üçün əlavə texniki xidmət və idarəetmə tələb edə bilər.

Xüsusi layihə tələblərinizə, komanda təcrübəsinə və istədiyiniz çeviklik və genişlənmə səviyyəsinə əsasən Selenium tətbiqini seçmək vacibdir. Selenium VebDriver geniş dil dəstəyi və çeviklik təklif edən ən çox istifadə edilən və çox yönlü proqramdır. Selenium IDE sürətli test yaratmaq və yeni başlayanlar üçün uyğundur, Selenium Grid isə paylanmış və paralel testin icrası üçün faydalıdır. [25]

2.11. Pythonda selenium tətbiqetmə üsulu

Python ilə Selenium brauzerlər və ya veb proqramlar üçün avtomatlaşdırılmış test işlərinin aparılması üçün istifadə olunur. Asanlıqla bir düyməyə toxunmaq, strukturlara məzmun daxil etmək, bütün saytı gözdən keçirmək və s. kimi testləri simulyasiya etmək üçün istifadə etmək olar.

Python yüksək səviyyəli, ümumi təyinatlı proqramlaşdırma dilidir. Onun dizayn fəlsəfəsi, kənar qayda vasitəsilə əhəmiyyətli boşluqların istifadəsi ilə kodun oxunuşunu vurğulayır.

2.12. Tətbiqetmə metodları

Obyektlər üzrə metodlar obyektin sinfinə qoşulmuş funksiyalardır; sintaksis normal metodlar və funksiyalar üçün sintaktik şəkərdir. Python metodları bəzi digər obyekt yönümlü proqramlaşdırma dillərində (məsələn, C++, Java, Objective-C, Ruby) gizlilikdən fərqli olaraq nümunə məlumatlarına daxil olmaq üçün açıq parametrə malikdir. Python həmçinin tez-tez dunder metodları adlanan üsulları təmin edirinstance.method(argument)Class.method(instance, argument)selfthis(adları cüt alt xətt ilə başlayan və bitən adlarına görə) istifadəçi tərəfindən müəyyən edilmiş siniflərə uzunluq, müqayisə, arifmetik əməliyyatlar və tip çevrilməsi daxil olmaqla, yerli əməliyyatlarla necə idarə olunduğunu dəyişməyə imkan verir.

2.13. Proqramlaşdırma nümunələri

Salam dünya proqramı:

```
çap ( 'Salam, dünya!' )
```

Müsbət tam ədədin faktorialını hesablamaq üçün proqram :

```
n = int ( input ( 'Rəqəm yazın və onun faktorialı çap olunacaq: ' ))
n < 0 olarsa :
   ValueError'u yüksəlt ( 'Mənfi olmayan tam ədəd daxil etməlisiniz' )
faktorial = 1
diapazondakı i üçün ( 2 , n + 1 ) :
```

```
    faktorial *= i
çap ( faktorial )
```

Python hazırda ən populyar və tələb olunan proqramlaşdırma dillərindən biridir. Səbəbi aydındır; istifadəçilərinə çoxlu üstünlüklər təklif edir və hərtərəfli hesab olunur. Flask, Django və s. kimi alətlərdən istifadə etməklə Python- dan istifadə edərək veb saytlar yarada bilərsiniz . Seleniumdan istifadə edərək veb avtomatlaşdırmanı həyata keçirmək də mümkündür. Siz Python-dan Data Science və Machine Learning və s. üçün istifadə edə bilərsiniz. Onun sadə sintaksisi və təqdim etdiyi tonlarla kitabxana və paketlər onu proqramçılar arasında sevimli halına gətirir.

Python öyrənmək inanılmaz dərəcədə asandır. O, açıq mənbəlidir, istifadəsi pulsuzdur, yüksək səviyyəlidir, şərh olunur və bütün dünyada mövcud olan hər hansı problemi həll etməyə hazır olan ən böyük tərtibatçı icmalarından birinə malikdir.

Python test üçün çoxlu daxili çərçivələrə malikdir ki, bu da sizə sazlama işləri aparmağa və sürətli iş axınları yaratmağa imkan verir. Selenium, Splinter kimi Python modulları və digər cross-brauzer və ya Robot, PyTest və s. kimi çarpaz platforma çərçivələri bu işi çox asanlaşdırır.

2.14. Seleniumun Python ilə bağlanması

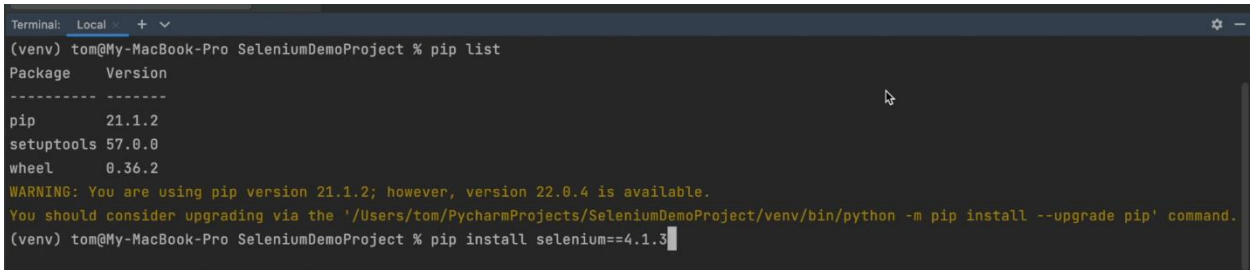
Python ilə Selenium üzərində işləməyə başlamaq üçün ilk addım, Selenium veb sürücüsündən istifadə edərək funksional test hadisələri yazmağınızdır. Daha sonra, brauzerlərdə sınaq işlərini avtomatik yerinə yetirəcək, arxa tərəfdə oturan Selenium serverinə sorğu göndərməlisiniz. Siz Firefox, Chrome, IE və s. daxil olmaqla istənilən brauzerdə testlər keçirə bilərsiniz. Dəyişən yeganə şey hər bir brauzer üçün xüsusi olan veb drayverlərdir.

Selenium-u Python ilə bağlamaq adətən API-lər vasitəsilə həyata keçirilir - onlardan Selenium veb sürücüsünün köməyi ilə funksional və ya qəbul testlərini yazmaq üçün istifadə edə bilərsiniz. Bu API istifadə edərək, siz intuitiv şəkildə bütün müxtəlif növ funksiyalara asanlıqla daxil ola bilərsiniz.

III FƏSİL. SELENİUM İLƏ AVTOMATLAŞDIRMA TESTİ

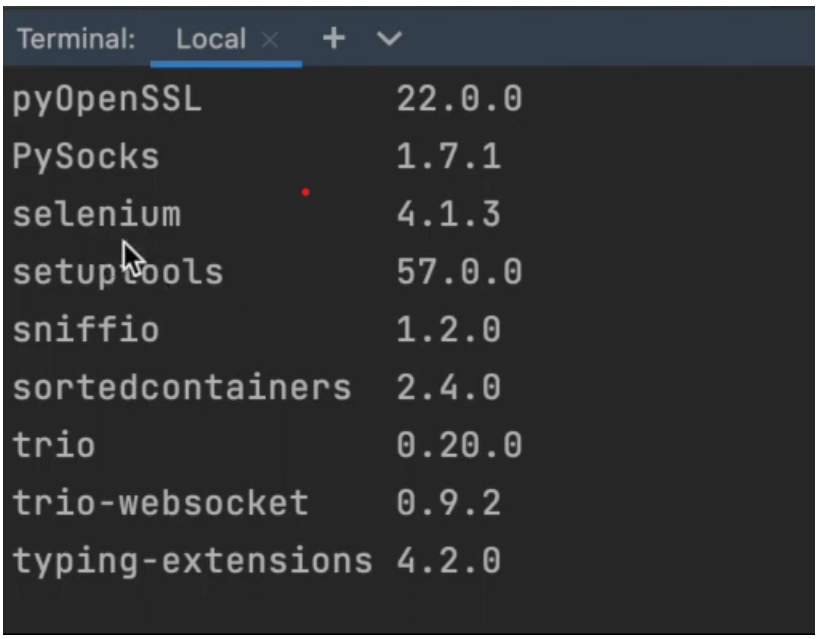
3.1. Seleniumun yüklənməsi

İstifadə edilən Pycharm vasitəsilə lazım olan selenium versiyasını terminal hissəsində “pip install selenium” commandı vasitəsilə yükləmək lazımdır.



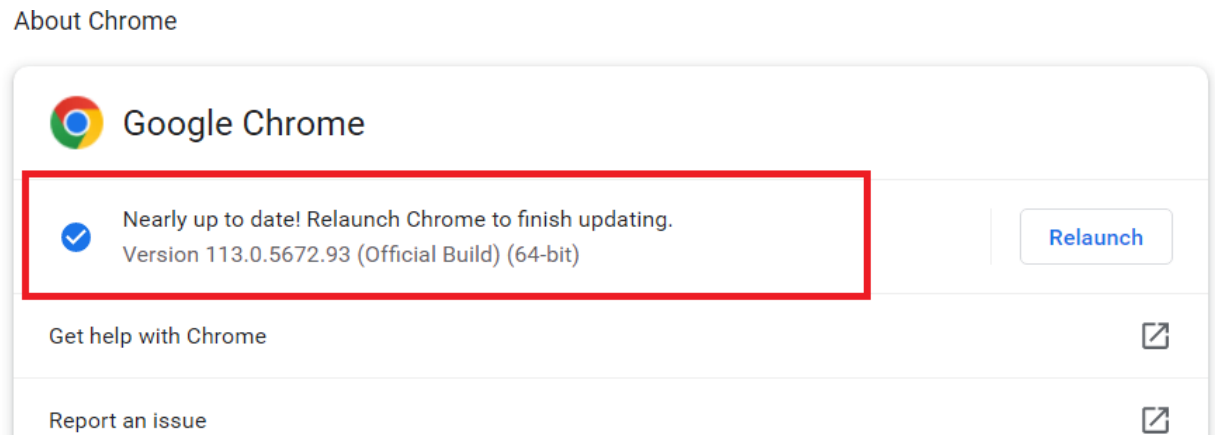
Şək 3.1.1. Selenium veraiyasının yüklənməsi

Yükləndikdən sonra “pip list” commandı vasitəsilə selenium-un yükləndiyini yoxlamaq olar.

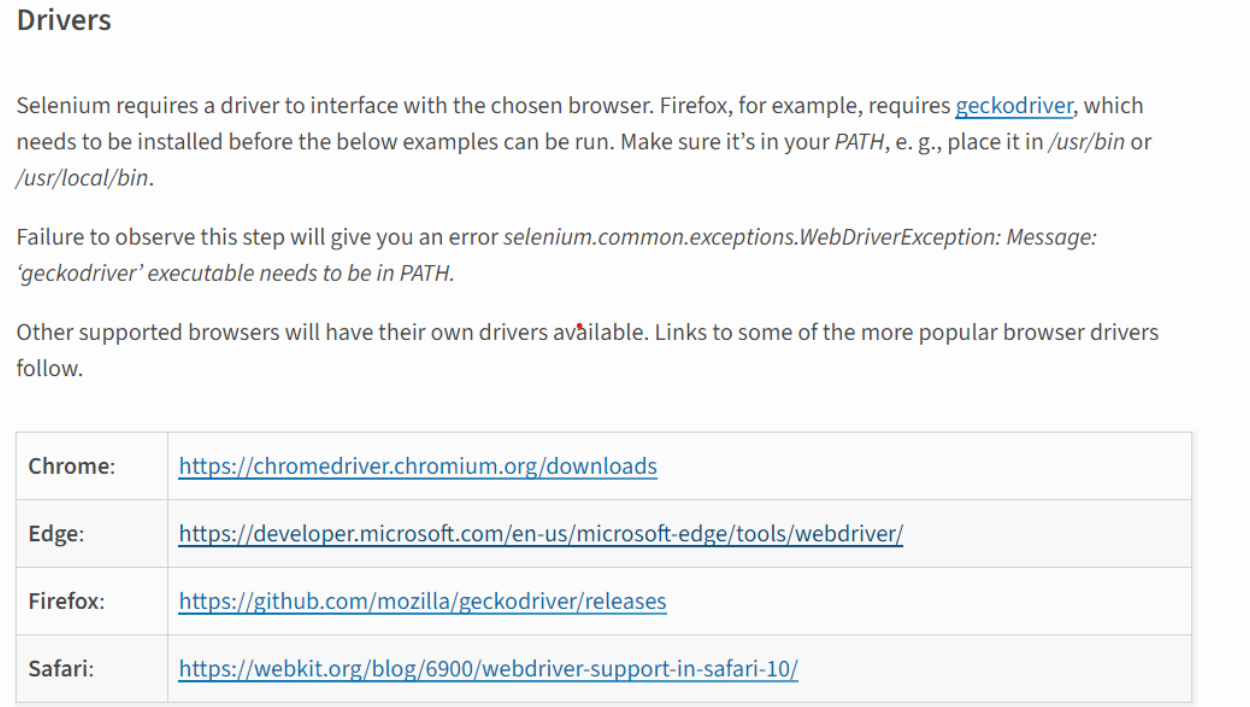


Şək 3.1.2. Selenium veraiyasının yoxlanması

Bir sonraki addımda seleniumda seleniumda test prosesini aparmaq üçün işlənildiyi browser üçün selenium driveri yükləmək lazımdır. Yükləməm zamanı Chrome browserin “about Google Chrome” hissəsidən versiyasını öyrənmək olar.



Şək 3.1.3. Brauserin seçilməsi



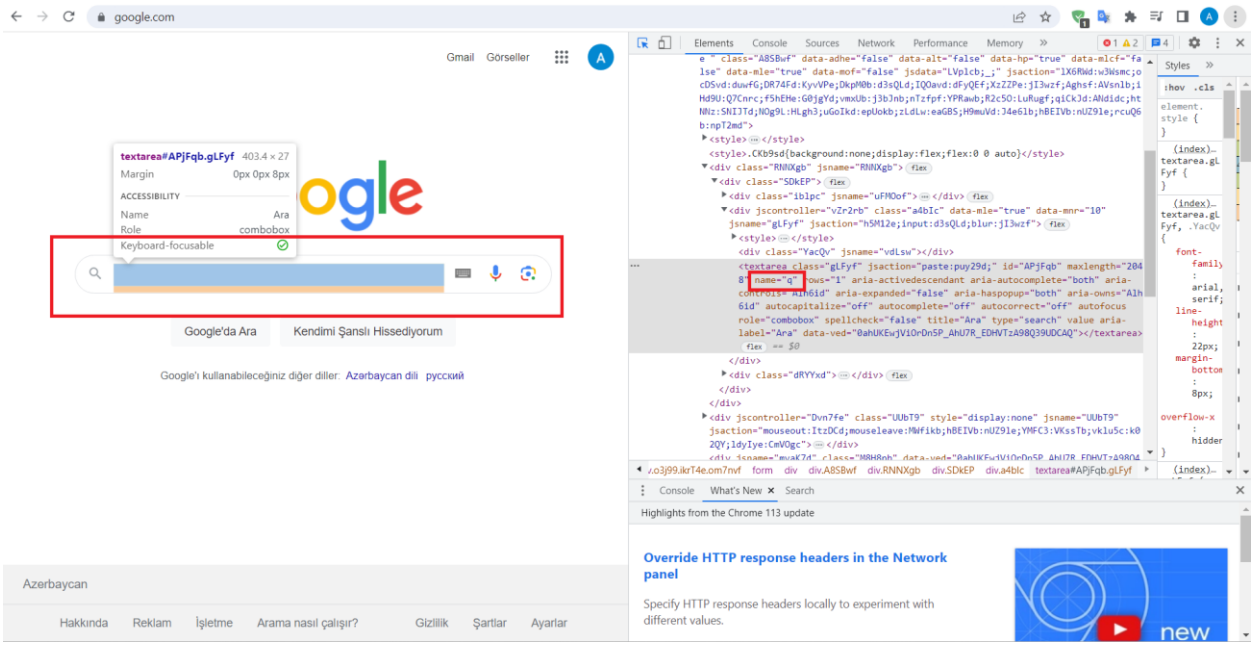
Şək 3.1.4. Brauserdən istifadə

Şək 3.1.5. Seleniumun yüklənməsi

Artıq selenium veb testetmə üçün hazırdır.

3.2 Veb testetmə with Selenium

Selenium vasitəsilə sadə testetməyə aid bu nümunədə google vasitəsilə istənilən məlumatın axtarışını avtomatlaşdırır. İlk addım olaraq command-da istifadə edilən (ID, NAME və s.) kimi dəyərləri öyrənmək üçün axtarış edilən sayt üzərində “inspect” vasitəsilə HTML ID ni öyrənib və daha sonra sadə axtarış edilən yazını automation vasitəsilə axtarış hissəsinə yazdırmaq lazımdır.

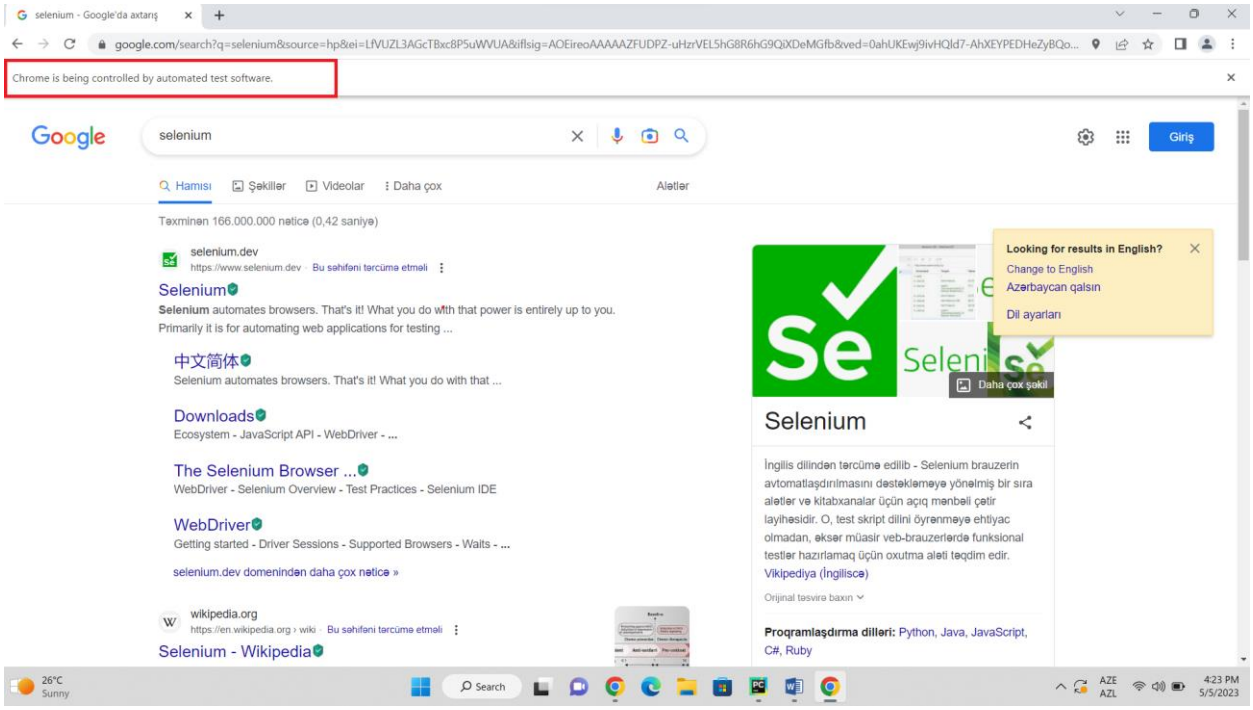


Şək 3.2.1. Google-da search-in name-nin götürülməsi

Fərqləndirilmiş hissədə gördüyünüz kimi axtarış avtomatlaşdırılmış test software vasitəsilə aparılır. Burada inspectdən google yazısının İD-i götürürük. İD 3növdə verilə bilər: Name ilə, id ilə və ya css selector vasitəsilə. Növbəti mərhələdə search hissədə axtarılacaq olan söz koda yazılır. Nümunədə Selenium sözü axtarılır. Axtarış icrası üçün axtar vurulmasının əvəz edən kod 14-cü sətirdə qeyd edilib. İnspectdən yenə name götürülür və koda yazılır. Burada CSS Selectorla qeyd edilib. Yazılan click isə enter üçündür. Daha sonra run verəndə proses işə düşür. Axtarış prosesi avtomatik icra olunur.

```python
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By


service = Service("./chromedriver.exe")
driver = webdriver.Chrome(service=service)
driver.get("http://www.google.com")
driver.maximize_window()
search = driver.find_element(By.NAME, "q")
search.send_keys("selenium")
driver.find_element(By.CSS_SELECTOR, "div.FPdoLc input.gNO89b").click()
```

Şək 3.2.2. Seleniumda kitabxana yaradılması



Şək 3.2.3. Seleniumla axtarışın avtomatlaşdırılması

Seleniumla mətn oxunma testinin aparılması:

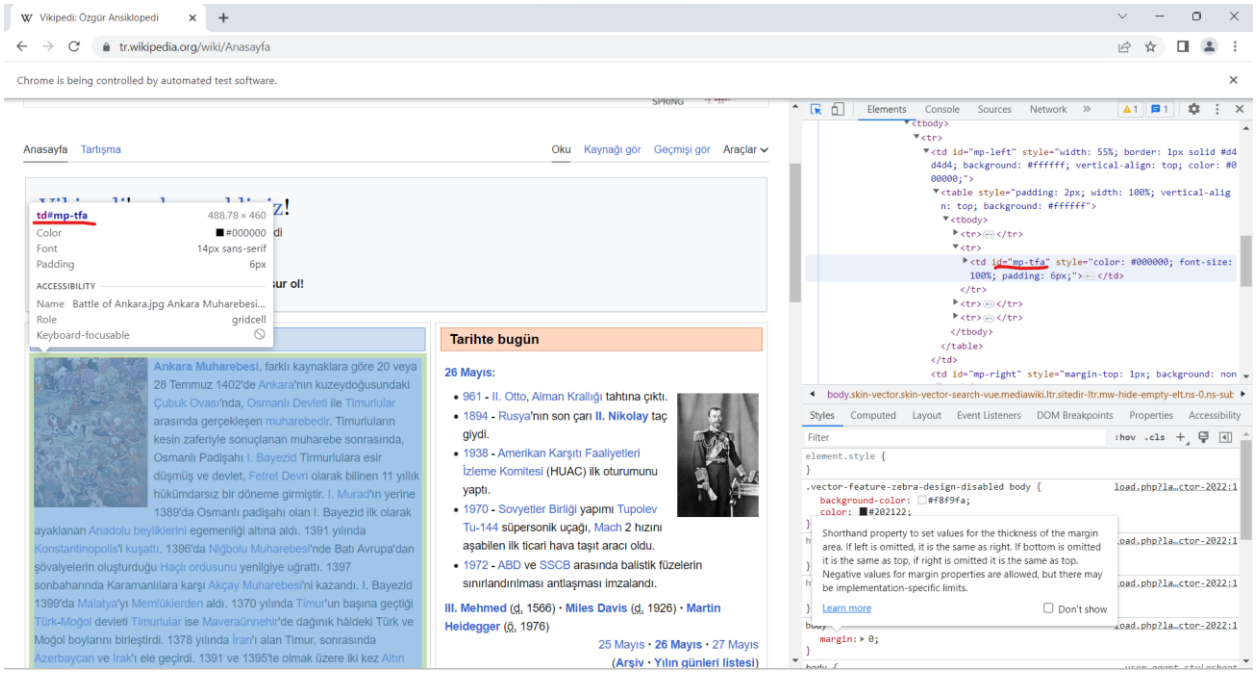
Aşağıda göstərilmiş nümunədə isə hər hansı veb səhifələrdə yerləşdirilmiş mətnin oxunulmasını selenium vasitəsilə test aparılır. Nümunə olaraq wikepediyanın gündəlik xəbələri göstərilmişdir. Bu nümunədə də mətnin oxunması üçün ilk öncə

gedilən veb səhifənin url-ni “driver.get()” funksiyasının köməyi ilə daxil etmək lazımdır.

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By


service = Service("./chromedriver.exe")
driver = webdriver.Chrome(service=service)
driver.get("https://tr.wikipedia.org/wiki/Anasayfa")
driver.maximize_window()
```

Şək 3.2.4. Seleniumda axtarış avtomatlaşdırılması kodlarının qeyd edilməsi

Daha sonra isə veb səhifə üzrə “inspect” –in köməyi ilə uyğun ID-ləri selenium koda daxil etmək lazımdır.



Şək 3.2.5. Seleniumda axtarışın avtomatlaşdırılmasının əks olunması

Axtarış üçün lazım olan məlumat yazılır və nəticə üçün run edilir. Nəticədə misalda əks olunduğu kimi vikipediyada “seçilmiş məqalə” sözü yazılır. Nəticə ekranda əks olunur.

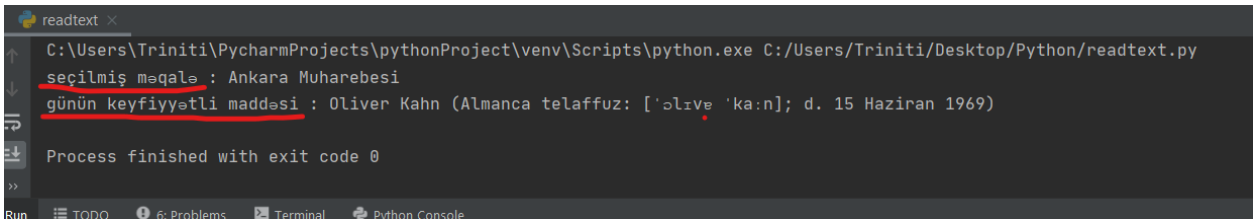
```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By


service = Service("./chromedriver.exe")
driver = webdriver.Chrome(service=service)
driver.get("https://tr.wikipedia.org/wiki/Anasayfa")
driver.maximize_window()
seçilmiş_məqalə = driver.find_element(By.ID, "mp-tfa").text
seçilmiş_məqalə = seçilmiş_məqalə.split(",")[0]
print("seçilmiş məqalə : " + seçilmiş_məqalə)
günün_keyfiyyətli_maddəsi = driver.find_element(By.ID, "mf-tfp").text
günün_keyfiyyətli_maddəsi = günün_keyfiyyətli_maddəsi.split(",")[0]
print("günün keyfiyyətli maddəsi : "+ günün_keyfiyyətli_maddəsi)
```

Şəkil 3.2.6. Axtaracağı mətnin başlığının qeyd edilməsi

Sonda nəticə olaraq testimizin düzgün işləyib işləmədiyini görmək üçün textin yazısı –nı “print” funksiyası vasitəsilə terminalımıza yazdırmaq lazımdır.

Şək 3.2.7. Axtarışın nəticəsi

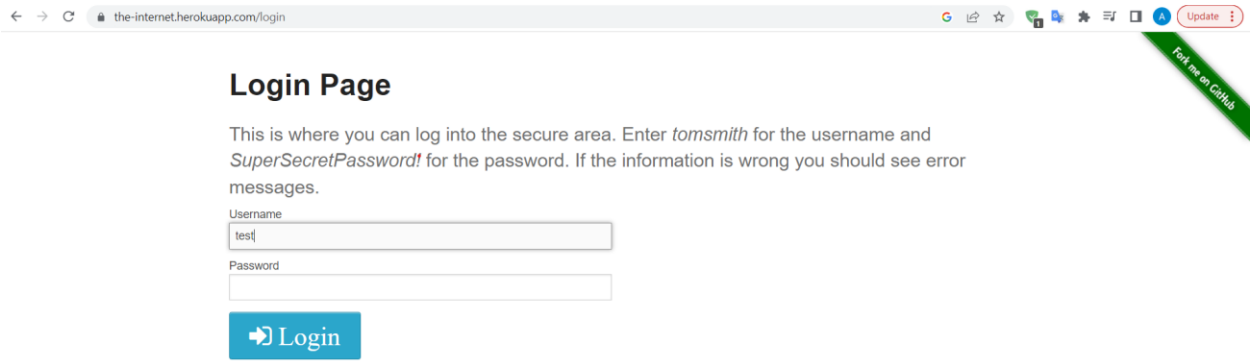
Şək 3.2.8. Axtarışın nəticəsinin növbəti mərhələsi



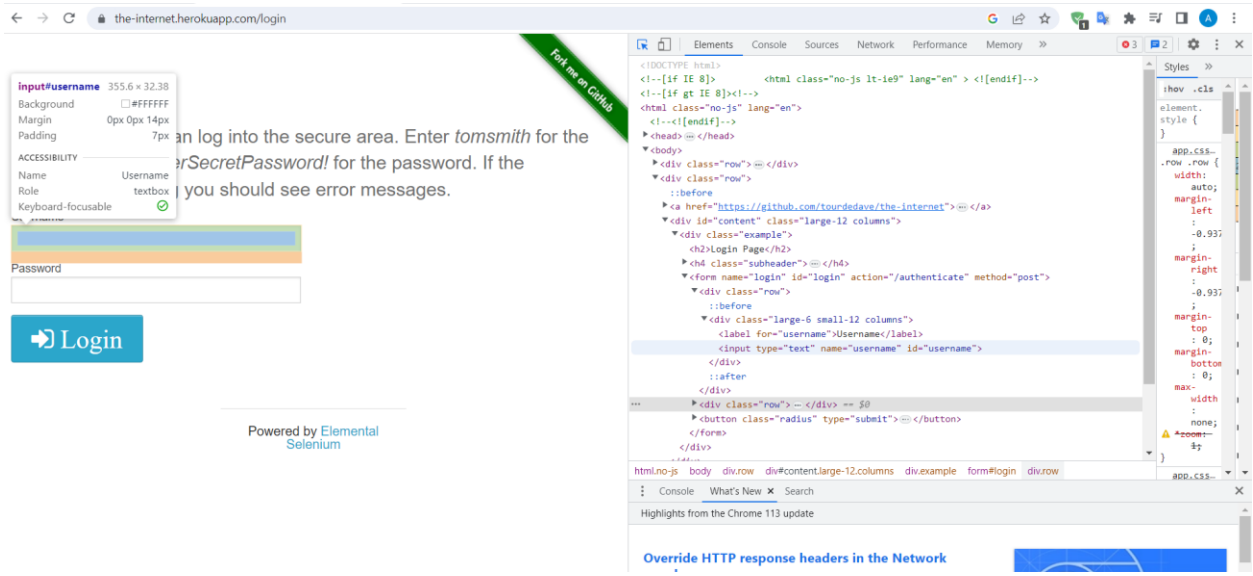
Şək 3.2.9. Kodların nəticəsilə uyğunluğunun təsdiqi

3.3 Login testinin aparılması

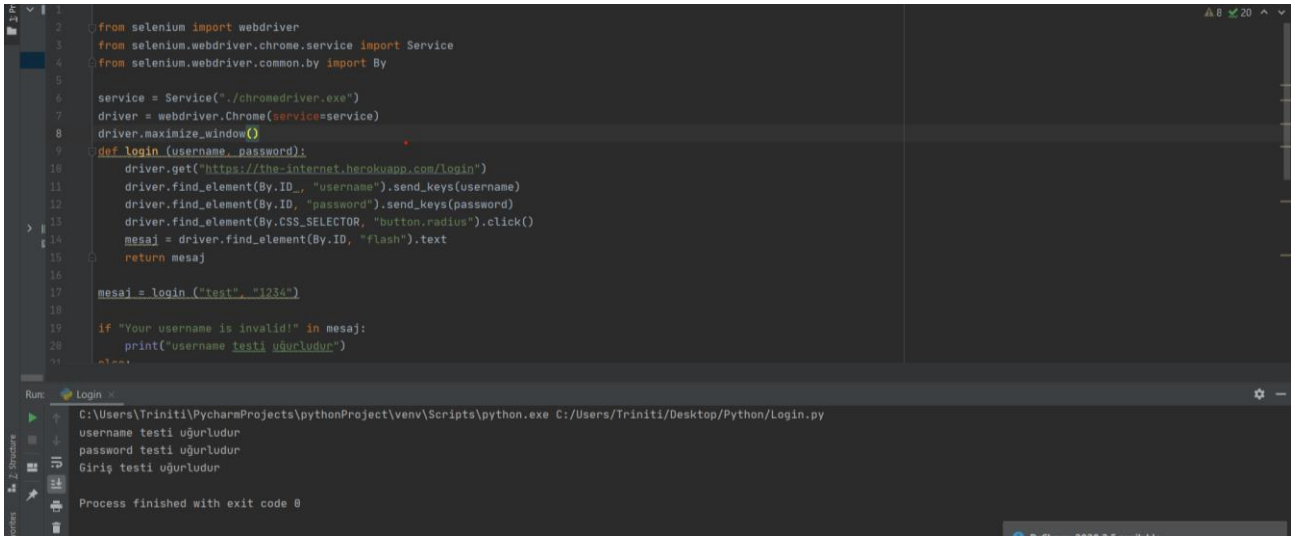
Göstərilmiş nümunədə test üçün https://the-internet.herokuapp.com/login veb səhifəsindən istifadə olunmuşdur. Testin aparılması zamanı uyğun ID-ləri funksiyaya daxil etdikdən sonra selenium vasitəsilə username və password-u daxil etmək lazımdır.



Şək 3.3.1. Loginin avtomatlaşdırılması

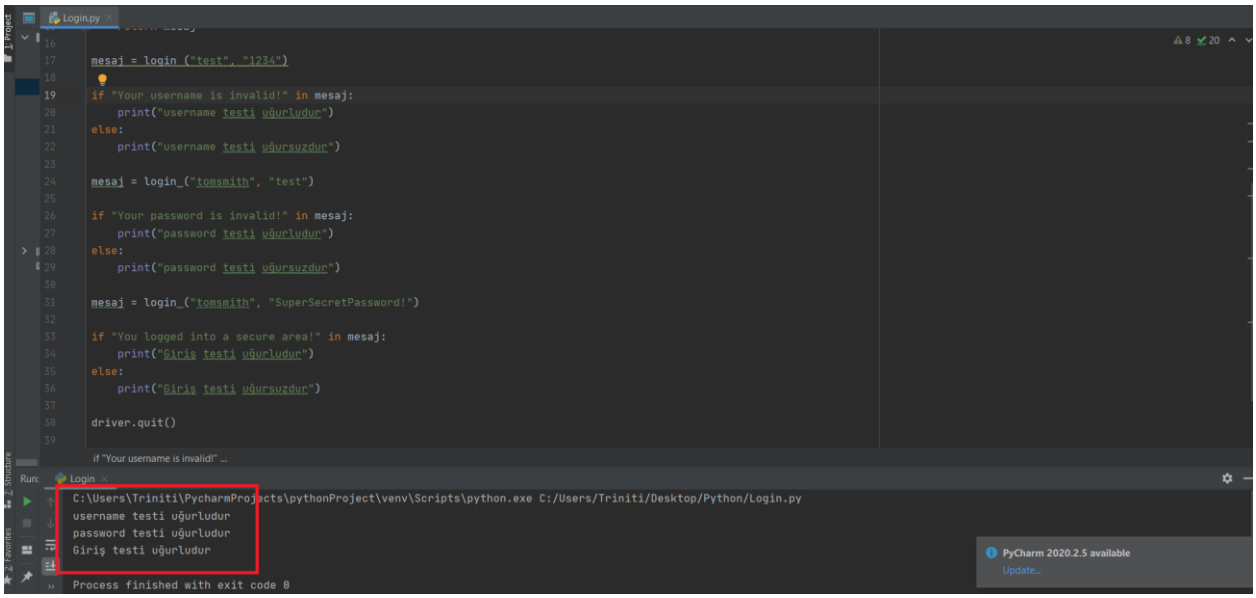


Şək 3.3.2. İnsertdən istifadə

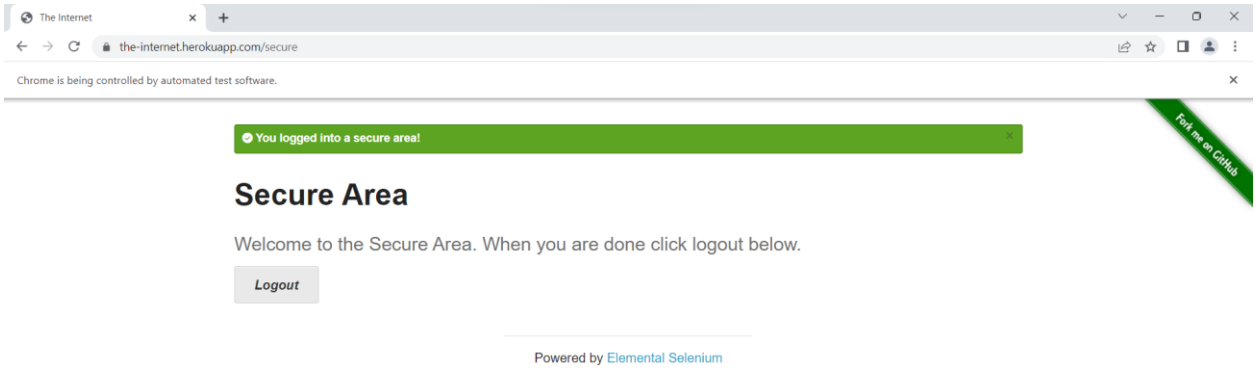


Şək 3.3.3. Kodun istifadəsinin yoxlanması

Bu hissədə run verilən zaman nəticə yazılan kodu əks etdirib-etdirməməsini yoxlamaq üşündür. Misalda qeyd edilən kimi bir neçə axtarış nümunəsi göstərilib ki, məsələn “Selenium”, “seçilmiş məqalə” və başqa axtarış nəticələri qeyd edilib, nəticəsi isə ekranda run hissədə və brauserdə əks edilmişdir. Nəticə qeyd olunması isə run bölməsində öz əksini tapmışdır. Bu “username testi uğurludur”, “password testi uğurludur”, “giriş testi uğurludur” sözləri ilə nəticənin uğurlu olmasını təsdiq edir.



Şək 3.3.4. Nəticənin qeyd olunması



Şək 3.3.5. Məqsədə çatıldı

Log in prosesi, username, password, giriş, və axtarış uğurla yerinə yetirilmişdir.

Nəticə

Bu tezisin əsas məqsədi son istifadəçinin onun funksionallığını yoxlamaq üçün edə biləcəyi kimi brauzerə girişi avtomatlaşdırmaq idi. Testin dizaynına görə, yerinə yetirilən nəticə faktiki işlə eyni olduğu üçün məqsədə çatıldı. Tətbiq dəyişdirilərsə, mənbə kodu daha sonra istifadə edilə bilər. Yəqin ki, tətbiqin tələbləri əsasında onun funksionallığını yoxlamaq üçün kiçik bir dəyişiklik tələb olunur. Əgər hər hansı bir halda o, VebElementi tapmırsa, element üzərində hərəkətləri yerinə yetirmək və ya bəzi digər alətlərdən istifadə etmək üçün brauzerlərin konsolundan elementi yoxlamaq lazımdır.

Bu tezis dinamik veb tətbiqinin funksionallığını yoxlamaq üçün necə avtomatlaşdırıla biləcəyini nümayiş etdirdi. Kifayət qədər asan görünsə də, avtomatlaşdırmaq üçün çox iş tələb olunur ki, bu da onu daha da çətinləşdirir. Selenium VebDriver-in öz dili yoxdur. Bu, digər proqramlaşdırma dillərindən asılıdır.

İstifadə olunmuş ədəbiyyat siyahısı

1. Alhaddad, M., & Dehlen, V. (2018). Quality Assurance of Autonomous Systems: State-of-the-Art Review and Challenges. In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 4011-4016). IEEE.

2. Alkhalaf, S., & Glinz, M. (2019). Testing AI-based Systems: A Survey on Challenges and Best Practices. In 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW) (pp. 42-49). IEEE.

3. Alshahwan, N., & Alhajj, R. (2015). Selenium testing tool. In Computational Intelligence in Data Mining-Volume 2 (pp. 183-201). Springer.

4. Baye, Y., Drame, A., & Konate, A. (2018). Test Automation of Web Applications: An Empirical Study on Selenium WebDriver. In 2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP) (pp. 1-6). IEEE.

5. Bhatt, D. V., Joshi, S., & Shah, S. (2018). Comparative analysis of automated web testing tools: Selenium, Coded UI, and QuickTest Professional. In 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN) (pp. 235-239). IEEE.

6. Biro, M., & Andler, S. (2018). Quality Assurance Automation: The Application of Machine Learning and AI Algorithms in Software Testing. In 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP) (pp. 345-352). IEEE.

7. Chandra, P., & Kumar, A. (2017). Automation Testing in Quality Assurance: A Review. International Journal of Innovative Research in Computer Science & Technology, 5(1), 63-68.

8. Smite, D., Wohlin, C., & Gorschek, T. (2016). Challenges and practices in aligning requirements with verification and validation: A case study of six companies. Empirical Software Engineering, 21(4), 1539-1577.

9. Gooch, D. (2017). Selenium Testing Tools Cookbook: Over 90 recipes to help you build and run automated tests for your web applications with Selenium WebDriver. Packt Publishing.

10. Hambleton, R., & Hambleton, M. (2017). Selenium Webdriver in Java: Learn with Examples. CreateSpace Independent Publishing Platform.

11. Hamza, S. (2018). Automated web testing using Selenium WebDriver and TestNG. In Proceedings of the 2018 International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (pp. 1-5). IEEE.

12. Kulkarni, V., & Prasad, A. (2017). Automation Testing in Software Quality Assurance. In Proceedings of the 2017 International Conference on Data Management, Analytics and Innovation (pp. 331-337). ACM.

13. Kumar, S., & Gupta, A. (2017). Automating web applications testing using Selenium and Java. In 2017 2nd International Conference for Convergence in Technology (pp. 1-6). IEEE.

14. Kazimierczak, D. (2018). Test Automation Using Selenium WebDriver with Java: Step by Step Guide. Independently published.

15. Keser, M., & Baris, E. (2019). Automated Web Testing with Selenium and WebDriver Using Java. Wiley.

16. Mathur, S., Goyal, S., & Pathak, R. (2020). An Empirical Study on Automation Testing in Quality Assurance. International Journal of Advanced Research in Computer Science, 11(4), 316-320.

17. Meenakshi, A., & Shobha, G. (2016). An effective approach for web application testing using Selenium WebDriver. In 2016 International Conference on Inventive Computation Technologies (pp. 1-4). IEEE.

18. Mittal, A. (2017). Selenium Framework Design in Data-Driven Testing: Build scalable frameworks and implement automated testing with Selenium WebDriver. Apress.

19. Nair, U. (2018). Selenium WebDriver: From Foundations To Framework. Packt Publishing.

20. Pandey, A., & Gupta, A. (2019). A systematic review of challenges and approaches in web application testing using Selenium WebDriver. International Journal of Advanced Research in Computer Science, 10(2), 8

21. Reddy, G. (2016). Selenium Design Patterns and Best Practices. Packt Publishing.

22. Shet, N. (2017). Mastering Selenium WebDriver 3.0: Boost the performance and reliability of your automated checks by mastering Selenium WebDriver, 2nd Edition. Packt Publishing.

23. Saleem, M. A., Ali, A., & Ali, A. (2021). Test automation of web applications: Comparative analysis of Selenium WebDriver and Appium. International Journal of Software Engineering and Computer Systems, 6(2), 69-78.

24. Shu, F., Xu, B., Zeng, Y., & Xiong, H. (2016). Evaluation of automated web testing tools: Selenium, Sahi and Watir. Journal of Electrical Engineering and Automation, 8(3), 70-77.

25. Silva, A. C., & Marques, A. (2018). Selenium WebDriver Recipes in C#: Second Edition. Apress.

26. Stahlhut, M., & Mesbah, A. (2017). Using machine learning to automate the extraction of interaction-based web vulnerabilities in rich internet applications. Empirical Software Engineering, 22(4), 1725-1762.

27. Sudharani, N., & Raju, C. (2018). Challenges and Approaches in Automation Testing. International Journal of Computer Science and Mobile Computing, 7(8), 271-276

28. Verma, A. (2019). Selenium Testing Tools Cookbook: Build automated testing frameworks with Selenium WebDriver using Java. Packt Publishing.

XÜLASƏ

Bu tezis Selenium çərçivəsindən istifadə edərək veb proqramların sınaqdan keçirilməsinə dair hərtərəfli tədqiqatı təqdim edir. Tədqiqat nəzəri əsasları, praktiki tətbiqi və Seleniumun Python proqramlaşdırma dili ilə inteqrasiyasını ətraflı başa düşməyi təmin etməyə yönəlmişdir. Tədqiqat veb tətbiqetmələrin sınaqdan keçirilməsində qarşıya çıxan çətinlikləri həll edir və avtomatlaşdırılmış və əllə sınaq yanaşmalarının üstünlüklərini və mənfi cəhətlərini araşdırır.1-ci fəsil veb tətbiqi testləri, o cümlədən onun tərifi və cəlb olunan problemlər haqqında məlumat təqdim edir. 2-ci fəsil proqram təminatının sınaqdan keçirilməsinin əsaslarını və avtomatlaşdırılmış sınaq konsepsiyalarını əhatə edən nəzəri çərçivəni yaradır. O, Selenium WebDriver və Selenium IDE daxil olmaqla, veb proqramların sınaqdan keçirilməsi üçün güclü bir vasitə kimi Seleniumun müxtəlif aspektlərini araşdırır. Seleniumun üstünlükləri, mənfi cəhətləri və tətbiqi davranışı və sistem tələbləri ilə birlikdə hərtərəfli müzakirə olunur. Bundan əlavə, Seleniumun Python proqramlaşdırma dili ilə inteqrasiyası araşdırılır.3-cü fəsildə veb proqramların sınaqdan keçirilməsində Seleniumun tətbiqini nümayiş etdirmək üçün praktiki nümunələr təqdim olunur. Bu nümunələrə Seleniumdan istifadə edərək mətn oxuma testinin aparılması və giriş testi ssenarilərinin aparılması daxildir. Bu dissertasiyada təqdim olunan tapıntılar və nümunələr veb proqram sistemlərinin keyfiyyətini və etibarlılığını artıraraq, veb tətbiqetmələrin sınaqdan keçirilməsi üsullarının inkişafı və başa düşülməsinə kömək edir.

SUMMARY

This dissertation presents a comprehensive study of web application testing using the Selenium framework. The study aims to provide a detailed understanding of the theoretical foundations, practical application and integration of Selenium with the Python programming language. The study addresses the challenges faced in testing web applications and examines the advantages and disadvantages of automated and manual testing approaches. Chapter 1 provides an overview of web application testing, including its definition and the challenges involved. Chapter 2 establishes a theoretical framework covering the fundamentals of software testing and automated testing concepts. It explores various aspects of Selenium as a powerful tool for testing web applications, including Selenium WebDriver and Selenium IDE. Selenium's advantages, disadvantages, and implementation behavior and system requirements are thoroughly discussed. In addition, the integration of Selenium with the Python programming language is explored. Chapter 3 provides practical examples to demonstrate the application of Selenium in web application testing. These examples include performing text reading testing and performing penetration testing scenarios using Selenium. The findings and examples presented in this dissertation contribute to the development and understanding of web application testing techniques, improving the quality and reliability of web application systems.

РЕЗЮМЕ

Эта диссертация представляет собой всестороннее исследование тестирования веб-приложений с использованием среды Selenium. Исследование направлено на подробное понимание теоретических основ, практического применения и интеграции Selenium с языком программирования Python. В исследовании рассматриваются проблемы, возникающие при тестировании веб-приложений, и анализируются преимущества и недостатки автоматизированных и ручных подходов к тестированию.Глава 1 содержит обзор тестирования веб-приложений, включая его определение и связанные с ним проблемы. Глава 2 устанавливает теоретическую основу, охватывающую основы тестирования программного обеспечения и концепции автоматизированного тестирования. В нем рассматриваются различные аспекты Selenium как мощного инструмента для тестирования веб-приложений, включая Selenium WebDriver и Selenium IDE. Подробно обсуждаются преимущества и недостатки Selenium, поведение при реализации и системные требования. Кроме того, рассматривается интеграция Selenium с языком программирования Python.Глава 3 содержит практические примеры для демонстрации применения Selenium при тестировании веб-приложений. Эти примеры включают тестирование чтения текста и выполнение сценариев тестирования на проникновение с использованием Selenium. Выводы и примеры, представленные в этой диссертации, способствуют развитию и пониманию методов тестирования веб-приложений, повышению качества и надежности систем веб-приложений.