

Testarea aplicației software

Lucrare de laborator nr 3

Subiect : Testarea automată aplicații web

Testarea aplicațiilor web cu mai multe pagini, cum ar fi magazinele online, este o parte integrantă a procesului de dezvoltare pentru a asigura calitatea produsului și satisfacția utilizatorilor.

Scopul principal al testării este de a asigura calitatea înaltă a produsului. Acest proces vă permite să identificați și să remediați erorile, defectele și deficiențele în funcționalitatea unei aplicații web, asigurându-vă că funcționează fiabil și fără erori.

Testarea are ca scop, de asemenea, să se asigure că toate paginile aplicației funcționează în mod fiabil. Aceasta include verificarea faptului că elementele de bază sunt afișate corect, funcționalități precum căutarea și filtrarea produselor funcționează corect, precum și procesarea comenzilor și a plăților.

Securitatea aplicațiilor este, de asemenea, punctul central al testării. Verificarea vulnerabilităților, prevenirea injecțiilor SQL, cross-site scripturile și alte amenințări sunt toate aspecte importante ale testării de securitate

1. Instalare seleniu :

- Instalați Selenium WebDriver pentru limbajul de programare pe care intenționați să îl utilizați (de exemplu, Java, Python, C#).
- Descărcați driverul de browser cu care intenționați să interacționați (de exemplu, ChromeDriver , GeckoDriver pentru Firefox).

2. Configurarea proiectului:

- Creați un proiect de automatizare de testare.
- Adăugați biblioteca Selenium WebDriver în funcție de proiectul dvs.

3. Teste de scriere:

- Folosiți seleniu WebDriver API pentru scrierea testelor.
- Creați cazuri de testare care reproduc acțiunile tipice ale utilizatorului pe diferite pagini ale aplicației (de exemplu, căutarea unui produs, adăugarea unui produs în coș, verificarea).

4. Localizatori de elemente:

- Utilizați selectoare (selectorii CSS, XPath) pentru a găsi elemente web pe o pagină.
- Asigurați-vă că localizatorii sunt unici pentru fiecare element, pentru a evita conflictele atunci când apar modificări de aspect.

5. Așteptări:

- Introduceți așteptări în codul dvs. de testare pentru a gestiona acțiunile asincrone și încărcarea paginii.

- Așteptările vor ajuta la evitarea rezultatelor fals pozitive în teste din cauza încărcării incomplete a paginii.

6. Utilizarea modelului de obiecte de pagină :

- Separați logica de interacțiune a elementului web și logica de testare folosind modelul Page Object.
- Creați clase de obiecte de pagină pentru fiecare pagină din aplicație care vor stoca localizatori și metode de interacțiune cu elementele din pagină.

7. Executarea testelor:

- Scrieți scripturi pentru a rula teste automat.
- Puteți utiliza sisteme de compilare precum Maven , Gradle sau instrumente CI/CD pentru a rula teste automat.

8. Analiza rezultatelor:

- După efectuarea testelor, analizați rezultatele.
- Documentați erorile dacă apar și furnizați informații dezvoltatorilor.

Testarea aplicațiilor cu mai multe pagini necesită luarea în considerare a multor aspecte diferite ale unei aplicații date. Mai jos puteți lua în considerare punctele de bază ale proiectării testelor pentru o aplicație de magazin online .

1. Testarea funcționalității:

- **Pagina principala:**
 - Verificarea afisarii elementelor de baza (logo, link-uri de navigare, cautare).
 - Testarea funcționării meniului principal și a submeniurilor.
 - Verificarea funcționalității accesului rapid la secțiuni (promoții, articole noi etc.).
- **Pagini de categorii și produse:**
 - Asigurați-vă că produsele sunt afișate corect.
 - Verificați funcționarea filtrelor și sortarea mărfurilor.
 - Testarea butoanelor pentru adăugarea articolelor în coș.
- **Coș și comenzi:**
 - Verificarea adaugării/eliminării articolelor din cos.
 - Testarea funcționării cupoanelor și reducerilor.
 - Revizuirea procesului de plată până la confirmarea finală.
- **Contul personal al utilizatorului:**
 - Se verifică înregistrarea și autentificarea.
 - Testarea funcționalității pentru modificarea datelor cu caracter personal.
 - Verificarea istoricului comenzilor și a stărilor acestora.

2. Testare de compatibilitate:

- Testarea funcționării în diverse browsere (Chrome , Firefox, Safari, Edge).
- Testare pe diverse sisteme de operare (Windows, macOS , Linux).

3. Test de performanta:

- Estimarea timpului de încărcare a paginii.
- Testare pe o conexiune lentă la Internet.
- Verificarea receptivității interfeței cu un număr mare de utilizatori simultani.

4. Testare de securitate:

- Verificarea unui certificat SSL.
- Testarea vulnerabilităților (injecții SQL, atacuri XSS etc.).
- Verificarea securității tranzacțiilor de plată.

Progres

1. Pregătiți un nou proiect pentru a crea un grup de teste pentru aplicația <https://demoqa.com/books>
2. Elaborați o listă de teste pentru a verifica funcționalitatea
3. Elaborați o listă de teste pentru a verifica baza de date (utilizați API furnizate pe site)
4. Efectuați testarea folosind testele compilate (7 teste de funcționalitate, 3 teste de baze de date, 2 teste de securitate)
5. Colectați și interpretați rezultatele programului testerului (completați tabelele de testare).
6. Analizați criticitatea rezultatelor testelor și pregătiți o concluzie
7. Completează raportul de laborator.

La finalizarea lucrărilor, întocmește un raport care să includă numele, prenumele, grupul, subiectul de lucru , o scurtă descriere a sarcinii și informații despre aplicația testată. Salvăm rapoartele finalizate în format PDF nume în format Prenume_Nume_număr și grupa_Laborator _Nr .