Testarea automatizata – SauceDemo magazin online

Obiective:

* Crearea de teste/scenarii folosind sintaxa Gherkin BDD(Given/When/And/Then)
* Definirea pașilor cu metodele apelate pentru realizarea testului
* Validarea pașilor si rezultatelor prin aserțiuni
* Utilizarea a diferitor utilizatori care permit reproducerea defectelor prezente in aplicație

Cerințe:

* Descrierea testelor in fișier de tip .feature
* Crearea legăturii dintre fișierul cu scenarii si pașii descriși in clasa (denumirea fisierului cu pasii descrisi trebuie sa fie cu prefix **\_test** al denumirii fisierului cu scenarii) de ex: saucedemo.feature – fisierul cu scenarii

Saucedemo\_test.py – fisierul cu pasii descrisi

* Fisierul cu pasii descrisi trebuie sa fie in mapa cu denumire **steps**
* Utilizarea bibliotecilor Python precum **json** pentru citirea datelor din fișier si **webdriver** care oferă o interfață pentru a scrie și a rula scripturi de automatizare.
* Versiunea Python recomandată: 3.8 sau mai nouă.
* Folosirea Seleniului pentru testarea a web aplicații
* Folosirea Behave pentru a rula testele

*Planificare și Implementare Logică*

* Identificam care vor fi scenariile pentru testare
* Testam manual scenariile pentru ajustarea rezultatului si scopului dorit
* Descriem in metoda fiecare pas pe care il parcurge utilizatorul
* Definim obiectele necesare pentru lucrul in web aplicatie. Se poate de folosit Inspectorul din web browser: Settings->MoreTools->Developer Tools

A screenshot of a computer

Description automatically generated

*Setarea Mediului de Lucru*

* Inițializarea unui mediu virtual Python și instalarea bibliotecilor necesare: Selenium pentru testarea web aplicatiilor, Behave pentru rularea testelor in Python

*Dezvoltarea Logică*

* Identificarea obiectelor dupa **Id** si validarea textului folosind **.text**
* Pentru introducerea datelor in cimpuri folosim .send\_keys([valoare])
* Pentru validare folosim **assert** daca valorile sunt dupa asteptari

**Structura Proiectului**

SauceDemo\_pythonProject/

│

├───features/

│ ├───steps/

│ ├───saucedemo\_steps.py

| ├───saucedemo.feature

│

├───jsonLibrary/

│ ├───behaveResults.json

│ ├───personalData.json

|

├───main/

| |───runners/

│ ├───runner\_behave.sh

│ ├───runner\_behave\_to\_json.sh

│ ├───runner\_streamlit\_app.sh

│ ├───app.py

│ ├───script.py

|

├───src/

│ ├───pages/

│ |───BasePage.py

│ |───SauceDemoPage.py

| ├───utility/

│ ├───Constants.py

│ |───Webdriver.py

└─────────────────────────

* app.py: Punctul de intrare al aplicației, care rulează interfața Streamlit.

Aici pornim suproces care ruleaza testele si citim datele din json fisier format dupa rulare si afidam datele pe pagina streamlit

* script.py: Aici stocam instructiunea de pornirea a subprocesului de rulare a testelor
* http://saucedemo\_steps.py: Implementarea pasilor scenariilor.

**Instalarea Bibliotecilor**

Instalează bibliotecile necesare executând următoarea comandă în terminal:

pip install behave

pip install streamlit

**Rularea scenariilor**

1. Configurarea pentru rulare a tuturor scenariilor folosind bash file cu behave instructiune

**A screenshot of a computer

Description automatically generated**

1. Pentru rularea a unui scenario specific, in terminal introducem instructiunea

Behave -n “Aici scrim denumirea scenariului”

1. Rulare folosind Streamlit

Configuram aplicatia sa ruleze din bash file folosind instructiunea

Streamlit run [Aici scrim calea spre app.py]

**Notiuni:**

**BDD** (Behavior Driven Development) - este o tehnică de dezvoltare software agilă și este ca o extensie a TDD, adică, Test Driven Development. În BDD, cazurile de testare sunt scrise într-un limbaj natural pe care chiar și non-programatorii îl pot citi (Given/When/And/Then)

**Selenium** este open-source tool de instrumente și biblioteci care este folosită pentru automatizarea browserului. Selenium permite utilizatorilor să-și testeze site-urile web în mod funcțional pe diferite browsere. **Selenium Webdriver** oferă o interfață pentru a scrie și a rula scripturi de automatizare. Fiecare browser are drivere diferite pentru a rula teste. Chrome are ChromeDriver, Firefox are GeckoDriver, Safari are SafariDriver și așa mai departe, datorită faptului că fiecare browser are un mod diferit de a efectua acțiuni, adică încărcarea unei pagini.

**Behave** este un cadru de dezvoltare bazată pe comportament (BDD) pentru Python, care permite echipelor să scrie teste ușor de citit în limbajul Gherkin. Oferă o abordare structurată a testării, asigurând o automatizare fiabilă și eficientă a testelor.