

Documentație Proiect Final

Proiectul este realizat utilizând "design pattern-ul" "**Page Object Model**".

Ca 'Framework' am folosit "**Selenium WebDriver**".

Testele sunt rulate folosind 'tool-ul' "**Maven**".

În clasa "**BasePage**" sunt prezente configurările pentru deschiderea unei instanțe de "Browser" la deschiderea fiecărui "test-case", urmând apoi, ca după rularea testului respectiv, acea instanță de "Browser" să se închidă. Rularea testelor o să se desfășoare în condiție "Full Screen", setarea fiind făcută la linia 26 din clasa menționată.

Fisierul "pom.xml" conține dependențele utilizate pentru întregul proiect, anume: "**Chrome Driver**", "**Selenium**", "**Maven**", "**TestNG**", "**GitHub**", "**Allure**", dar și "**Faker**".

"**Java Faker**", este o librărie folosită pentru a genera date fictive. Oferă diferite clase și metode pentru a produce o date cu aspect real care variază de la număr de telefon, adresă, e-mail genul de muzică, naționalitate și multe altele. Este foarte folositoare atunci când dorim să folosim un substituent, dar nu avem date reale.

Proiectul este format din două pachete ("Pages" & "TestCases"):

- "Pages": conține clasa "**BasePage**" care este responsabilă cu configurări și prerechizite necesare rulării testelor, dar și pentru "Browser". De asemenea, în "Pages" găsim și clasele "**RegisterAccountPage**", "**AddAddressPage**", "**BlogArticlePage**", "**EditYourAccountPage**", "**AddOnsFeaturedPage**", "**HomePage**". Aceste clase conțin locatorii cu elementele folosite pentru automatizarea testelor.
- "TestCases": conține clasele denumite "**RegisterAccountTests**", "**AddAddressTests**", "**BlogArticleTests**", "**EditYourAccountTests**", "**AddOnsFeaturedTests**", "**HomeTests**". În aceste clase sunt prezente "test-case"-urile pentru elementele din clasele aflate în pachetul "Pages".

Pentru a rula testele folosim comanda "**test**" care poate fi găsită accesând butonul 'm' din partea dreaptă-sus, sub Lifecycle. Dacă comanda "**test**" eșuează, este recomandat să rulăm comenzile "**clean**", "**validate**", "**compile**", "**test**" în ordine.

După ce testele au fost rulate, generăm un raport folosind comenzile **“allure:report”** si **“allure:serve”** aflate sub **“Plugins”/“allure”**. Comanda **“allure:report”** va genera raportul pe baza testelor rulate iar comanda **“allure:serve”** va afişa raportul.