Implementati o clasa Login care sa mosteneasca unittest.TestCase

Gasiti elementele in partea de sus folosind ce selectors doriti voi

setUp()

Driver

<https://the-internet.herokuapp.com/>

Click pe Form Authentication

tearDown()

Quit browser

Test1

Verificati ca noul url e corect

Test2

Verificati ca page title e corect

Test3

Verificati textul de pe elementul xpath=//h2 e corect

Test4

Verificati ca butonul de login este displayed

Test5

Verificati ca atributul href al linkului ‘Elemental Selenium’ e corect

Test6

Lasati goale user si pass

Click login

Verifica ca eroarea e displayed

Test7

Completeaza cu user si pass invalide

Click login

Verifica ca mesajul de pe eroare e corect

Este si un x pus acolo extra asa ca vom folosi solutia de mai jos

expected = 'Your username is invalid!'

self.assertTrue(expected in actual, 'Error message text is incorrect')

Test8

Lasati goale user si pass

Click login

Apasa x la eroare

Verifica ca eroarea a disparut

Test9

Ia ca o lista toate //label

Verifica textul ca textul de pe ele sa fie cel asteptat (Username si Password)

Aici e ok sa avem 2 asserturi

Test10

Completeaza cu user si pass valide

Click login

Verifica ca noul url CONTINE /secure

Foloseste un explicit wait pentru elementul cu clasa ’flash succes’

Verifica ca elementul cu clasa=’flash succes’ este displayed

Verifica ca mesajul de pe acest element CONTINE textul ‘secure area!’

Test11

Completeaza cu user si pass valide

Click login

Click logout

Verifica ca ai ajuns pe <https://the-internet.herokuapp.com/login>

BONUS

Test12 - brute force password hacking

Completeaza user tomsmith

Gaseste elementul //h4

Ia textu de pe el si fa split dupa spatiu. Considera fiecare cuvant ca o potentiala parola

Foloseste o structura iterativa prin care sa introduci rand pe rand parolele si sa apesi pe login

La final testul trebuie sa imi printeze fie

‘Nu am reusit sa gasesc parola’

‘Parola secreta este [parola]’