



Întâlnirea 12

Final Project

Sfaturi generale

- Să tratați cu **seriozitate** și **profesionalism** acest nou obiectiv.
- Cei care își ating obiectivele nu sunt întotdeauna cei mai smart, dar întotdeauna vor fi cei mai **muncitori!**
- Alocați-vă timp pentru studiu. Rutina dă **consistență**. Consistența dă **exelență**.
- Să faceți tot posibilul să participați la **toate** sesiunile live.
- Să vă lăsați **comentarii** explicative în cod. Notițe pentru voi din viitor.
- Recomand să vizualizați **înregistrarea**. Să vă notați aspectele importante + întrebări pentru trainer pentru ora următoare.
- Să vă faceți **temele** și unde nu reușiți singuri, să întrebați pe **grup**. Trainerul va **răspunde** și vor beneficia și ceilalți cursanți de răspuns.
- Puteți chiar să faceți un grup doar de studenți și să vă întâlniți o dată pe săptămână să discutați temele **împreună**. Fiecare va veni cu o perspectivă nouă și în final toți vor avea de câștigat.
- În timpul orelor, să aveți **curaj** să puneți **întrebări** când ceva nu e clar.

Reguli curs

- Va exista un sheet de prezență.
- În cadrul acestuia ne vom asuma și noțiunile învățate. Nu trecem mai departe până nu își asumă toți noile concepte.
- Temele se vor adauga în Folderul grupei, veți face fiecare folder cu numele vostru. Veți primi feedback la aceste teme.
- Temele vor fi împărțite în 2 categorii.
 - Obligatorii (se pot face doar cu noțiunile învățate la clasă)
 - Opționale (acestea vor fi mai advanced și necesită poate extra research). Acest lucru îi va motiva și pe cei care au mai mult timp și le place să se aventureze prin task-uri mai dificile.
- Vă rog să mă întrerupeți oricând aveți întrebări. Doar așa îmi pot da seama unde trebuie să mai insist cu explicații/exemple.
- Vă rog să intrați cu 3 minute mai devreme în caz că apar probleme tehnice. Astfel putem profita la maxim de cele 2 ore alocate.
- Dacă nu puteți intra, sau dacă întârziați, anunțați trainerul pe grup.

Obiective principale

Până la final TOȚI* veți avea:

- Cunoștințe solide despre bazele programării în Python
- Cunoștințe mai avansate și extrem de utile despre programarea bazată pe obiecte.
- Capacitatea să identifice elemente și să scrie test scripts cu ajutorul Selenium
- Un Proiect final de testare automată a aplicațiilor web.
 - Acesta va folosi tendințele actuale: metodologia Behavior Driven Development și Page Object Model Design pattern.
 - Vă avea capacitatea să genereze rapoarte HTML ('living documentation')
 - Veți ști de la A la Z acest framework, astfel că veți avea capacitatea să continuați să îl dezvoltați post curs acest proiect (pentru orice website doriți).
- Noțiuni de bază despre API testing. (testarea backend - ce e în spate la un website).

* toți cei care sunt activi, implicați, își fac temele, dedică timp pentru studiu individual și pun întrebări trainerului vor atinge aceste obiective.

Obiective secundare

Nu fac parte din curricula cursului LIVE dar vă punem la dispoziție materiale extra că să aveți un avantaj la interviuri. Sfatul meu e să vă focusați pe ele doar după cursul live. Să nu fiți overwhelmed de new info.

- Cunoștințe ale bazelor de date relaționale - mySQL (Curs baze de date)
- Cunoștințe teoretice despre testarea manuală - acces la o platformă mobilă
- Capacitatea de a construi un mic brand personal (Curs Portofoliu Wordpress). Trebuie să ai:
 - Website propriu prin care angajatorul să te cunoască pe tine și munca ta
 - CV european în eng / sau canva.com
 - Profil LinkedIn
 - Github public (un loc în cloud unde se pune codul scris de tine)
 - Veți primi feedback dacă ne trimiteți un email cu ele la hello@itfactory.ro

Obiective Întâlnire 12

- Să finalizăm proiectul final
- Să facem un test cap coadă, finalizat cu assert și raport html
- Să știm să parametrizăm testele și să folosim tabele de valori
- Să înțelegem structura proiectului astfel încât să putem scrie singuri teste

Steps Folder

- Conține 1:1 cate o pagină de pași pentru fiecare pagină din proiect
- Deasupra metodelor scriem scenariile cu sintaxa gherkin folosindu-ne de decoratorii aferenți (@given, @when, @then, @and, @but)

```
@given('login: user is on the login page')
def step_impl(context):
    context.login_page.navigate_to_jules()

# username password parameters are set in login.feature
@when('login: user enters valid username "{username}" and valid password "{password}"')
def step_impl(context, username, password):
    context.login_page.login(username, password)

@when('login: user clicks on forgot pass button')
def step_impl(context):
    context.login_page.click_forgot_pass_btn()

@then('login: verify that invalid credentials error is displayed')
def step_impl(context):
    context.login_page.verify_error_displayed()
```

Features folder

- Conține fișiere de tip .feature în care scriem scenariile de test

```
Feature: Jules login feature

  # se ruleaza inainte de fiecare test
  Background:
    Given login: user is on the login page

  @smoke
  Scenario: Login fara tabel de valori
    When login: user enters valid username "itfactory.ro@gmail.com" and valid password "pass123"
    Then login: verify that invalid credentials error is displayed

  @test
  Scenario Outline: Login cu tabel de valori
    When login: user enters valid username "<email>" and valid password "<pass>"
    Then login: verify that invalid credentials error is displayed

  Examples:
    | email | pass |
    | itfactory1.ro@gmail.com | password1 |
    | itfactory2.ro@gmail.com | password2 |
    | itfactory3.ro@gmail.com | password3 |
```


Features folder

- Aici ținem testele. Fiecare feature file testează o funcționalitate majoră a aplicației și poate conține mai multe scenarii de tests
- În Feature trecem titlul capabilității testate
- În Background trecem pașii care se execută înainte de fiecare test
- În Scenario: punem titlul testului
- Sub Scenario trecem pașii din test cu sintaxa gherkin
- Scenario Outline: ne permite să folosim un tabel de valori
- @smoke și @test sunt taguri care au rolul să împartă testele în categorii - aceste nume de taguri sunt la liberă alegere de către voi
- În Examples punem parametri pentru teste într-o structură tabelară
- Fiecare rând din tabel va fi un test run individual



Întrebări de interviu:

- De ce ar fi bine să avem o clasă BasePage în proiect?
- Ce conține o clasă de tip page?

Întrebări & curiozități?