# Selenium testy

## Test přihlášení (Login)

**Login-1 (Úspěšné přihlášení):**

Otevřít přihlašovací stránku.

Vyplnit formulář platnými uživatelskými údaji (uživatelské jméno/email a heslo).

Odeslat formulář.

Ověřit, že dojde k přesměrování na správnou cílovou stránku

**Login-2 (Neúspěšné přihlášení - neplatné údaje):**

Otevřít přihlašovací stránku.

Vyplnit formulář neplatnými údaji (špatné heslo nebo chybějící uživatelské jméno).

Odeslat formulář.

Ověřit, že se zobrazuje příslušné chybové hlášení a nedojde k přesměrování.

**Login-3 (Kontrola renderování formuláře):**

Otevřít přihlašovací stránku.

Ověřit, že všechny potřebné prvky (pole pro jméno/heslo, tlačítko pro odeslání) jsou na stránce zobrazeny.

## Test odhlášení (Logout)

**Logout-1 (Úspěšné odhlášení):**

Přihlásit se pomocí platných údajů (scénář úspěšného přihlášení).

Kliknout na tlačítko pro odhlášení

Ověřit, že dojde k přesměrování na přihlašovací stránku nebo jinou správnou URL.

Zkusit přistoupit na chráněnou stránku a ověřit, že přístup již není povolen (přesměrování na přihlášení).

## Test registrace

**SignUp-1 (Úspěšná registrace):**

Otevřít registrační stránku.

Vyplnit formulář platnými údaji.

Odeslat formulář.

Ověřit, že dojde k registraci nového uživatele a přesměrování (např. na přihlašovací stránku).

**SignUp-2 (Neúspěšná registrace - duplicitní email):**

Otevřít registrační stránku.

Vyplnit formulář s již existujícím emailem.

Odeslat formulář.

Ověřit, že se zobrazí chybová hláška a registrace není provedena.

**SignUp-3 (Neúspěšná registrace – nesouhlasí pass1 a pass2):**

Otevřít registrační stránku.

Vyplnit formulář s různými hesly.

Odeslat formulář.

Ověřit, že se zobrazí chybová hláška a registrace není provedena.

# Accounts testy

**UserListView**

**Testovací případ:**  
Přihlásíme se, zavoláme URL pro seznam uživatelů a ověříme response, šablona je správná a odpovídající data v kontextu.

**Hotovo 15.3.**

**UserDetailView**

**Testovací případy:**

Na základě existujícího ID uživatele ověříme funkčnost.

**Hotovo 15.3.**

Na neexistující ID ověříme, že systém vrací chybu.

**Hotovo 15.3.**

**UserUpdateView**

**Testovací případy:**

Validní POST požadavek zkontroluje aktualizaci dat a přesměrování na URL

**Hotovo 15.3.**

Negativní test – pokud není formulář validní nebo jsou špatně vyplněná data, kontrola, že formulář se znovu vykreslí s chybovým hlášením.

**Hotovo 15.3.**

**UserDeleteView**

**Testovací případy:**

Testování přístupu na stránku potvrzení smazání.

**Hotovo 15.3.**

Testování samotného smazání a následné kontrola, že objekt je odstraněn.

**Hotovo 15.3.**

**SignUpView**

**Testovací případy:**

Ověření základní funkčnosti formuláře pro registraci.

**Hotovo 15.3.**

Kontrola duplicitního e-mailu a zobrazení příslušné chybové zprávy v odpovědi.

**Hotovo 15.3.**

**CustomLoginView**

**Testovací případy:**

Kontrola, že při nesprávném heslu nebo jménu se neprovede přihlášení a uživatel je informován o problému.

**Hotovo 15.3.**

Ověření úspěšného přihlášení a následného přesměrování na URL.

**Hotovo 15.3.**

Form testy:

SignUpForm:

Zadávám platná neuexistující data. Test formuláře projde pozivně.

Hotovo 15.3.

Zadávám data s již existujícím mailem. Formulář invalidní.

Hotovo 15.3.

Formulář není validní, pokud chybí povinné pole (uživatel, jméno, příjmení)

Hotovo 15.3.

Formulář není validní, pokud se zadaná hesla neshodují

Hotovo 15.3.

UpdateForm:

Ověření, že formulář je validní při zadání platných dat   
a lze uložit změny

Hotovo 15.3.

Formulář není validní, pokud chybí povinné pole (username, first\_name, last\_name)

Hotovo 15.3.

Ověření, že e‑mail je volitelné pole – prázdný e‑mail nevyvolá chybu

Hotovo 15.3.

# Attendance testy

**Test redirect\_dashboard**

Testovat přesměrování z URL ("/") na URL dashboardu.

**Test dashboard\_view**

Ověřit, že dashboard stránky vrací úspěšný status

Kontrola použití správného šablonového souboru.

Zjistit, že kontext obsahuje očekávaná data

**Test main\_panel\_view**

Kontrolovat dostupnost hlavní panelové stránky

Ověřit správné použití šablony a přítomnost správných dat v kontextu.

**Test create\_employee\_get**

Otestovat GET požadavek na stránku pro vytvoření zaměstnance.

Ověřit, že je zobrazen formulář a načtení se správné šablona.

**Test create\_employee\_post\_valid**

Kontrola odchozí POST požadavek s validními daty.

Potvrzení, že po odeslání validních dat dojde k přesměrování a že se v databázi vytvoří nový záznam zaměstnance.

**Test create\_employee\_post\_invalid**

Ověření, že při odeslání nevalidních dat (chybějící povinné pole, nesprávný formát) se formulář znovu zobrazí se správnými chybovými hláškami.

Zůstává na stránce

**Test save\_vector\_to\_db\_post**

Simulovat POST požadavek na view pro uložení vektoru do databáze

Test view návrat. status a případně provede další kontroly dle implementované logiky.

# FaceVector testy

**Test vytvoření instance FaceVector:**

Vytvoreni instance ti m získat pristup k metodam a funkcim

**Test existence objektu FaceVector v databázi**

Zjistit vytovreni objectu a zdali ma v sobe nejakou hodnotu

**Test typové hodnoty atributů FaceVector**

Test jestli datovy typ odpovida

**Test metod šifrování a dešifrování vektorových dat**

Kontrola stavu před a po sifrovani a zdali porovnani je shodne

**Test finálního tvaru instance FaceVector po všech operacích**

Test stavu databaze, zejména hodnot v tabulkách zdali odpovidaji ocekavanym hodnotam