

LinkedIn Gönderi Paylaşma Testi

Selenium, linkedin.com adresine giriş yaparak sayfanın doğru görüntülenip görüntülenmediğini kontrol ediyor.

```
# Web sayfasının ChromeDrive ile açılması
self.driver.get("https://www.linkedin.com")
# Doğru URL'ye sahip sitenin yüklenip yüklenmediğini test etme
self.assertEqual(self.driver.title, "Log In or Sign Up")
```

Giriş sayfası açıldıktan sonra, sayfa üzerine email input alanı olup olmadığı kontrol ediliyor. Daha sonra verilen email değeri input alanı içerisine yazılıyor.

```
email_elem = self.driver.find_element(By.ID, 'session_key')
# Email input alanı erişilebilir mi kontrol et
self.assertTrue(email_elem.is_enabled())
# Email input alanını doldurma
email_elem.send_keys(self.email)
```

Giriş sayfası açıldıktan sonra, sayfa üzerine parola input alanı olup olmadığı kontrol ediliyor. Daha sonra verilen parola değeri input alanı içerisine yazılıyor.

```
password_elem = self.driver.find_element(By.ID, 'session_password')
# Password input alanı erişilebilir mi kontrol et
self.assertTrue(password_elem.is_enabled())
# Şifre input alanını doldurma
password_elem.send_keys(self.password)
```

Giriş sayfası üzerinde bulunan iki input alanı doldurulduktan sonra “GİRİŞ YAP” butonuna erişilip erişilmediği test ediliyor. Daha sonrasında input alanları dolu ise butona tıklanıyor.

```
login_button = self.driver.find_element(By.CSS_SELECTOR,
'.sign-in-form__submit-button')
# Giriş butonu görünür mü ve tıklanılabilir mi kontrol et
self.assertTrue(login_button.is_displayed())
self.assertTrue(login_button.is_enabled())
# Giriş butonuna tıklanılır
login_button.click()
```

Giriş başarılı olduktan sonra LinkedIn'in anasayfa'ya yönlendirilecektir. Ana sayfanın doğru görüntülüp görüntülenmediği kontrol ediliyor.

```
home_page = self.driver.find_element(By.ID, "voyager-feed")
# ana sayfa açıldı mı kontrol et
self.assertTrue(home_page.is_displayed())
```

Anasayfa içerisinde yer alan, gönderi paylaşmak için, butona görüntülenip görüntülenmediği ve daha sonrasında erişilip erişilmediği kontrol ediliyor. Eğer bir sorun yoksa butona tıklanıyor.

```
add_post_button =
self.driver.find_element(By.CSS_SELECTOR, '.artdeco-button.artdeco-button
--muted.artdeco-button--4.artdeco-button--tertiary.ember-view.share-box-
feed-entry__trigger')
# Gönderi paylaşma butonu görünür mü ve tıklanılabilir mi kontrol et
self.assertTrue(add_post_button.is_displayed())
self.assertTrue(add_post_button.is_enabled())
add_post_button.click();
```

Butona basıldıktan sonra LinkedIn sayfasında gönderi paylaşmak için içerisinde bir input alanı ve bir buton bulunan modal açılıyor. Modal'ın erişilip erişilmediği kontrol edildikten sonra belirlenen paylaşım metni input içerisine yazılıyor.

```
# post paylaşma
time.sleep(2)
self.postText = self.postText + str(random.randint(0,100000))
post_text_area =
self.driver.find_element(By.CSS_SELECTOR, ".ql-editor.ql-blank")
# editör alanı erişilebilir mi kontrol et
self.assertTrue(post_text_area.is_enabled())
# post yazısını editöre yazma işlemi
post_text_area.send_keys(self.postText)
```

Modal içerisinde yer alan butona görüntülenip görüntülenmediği ve daha sonrasında erişilip erişilmediği kontrol ediliyor. Eğer bir sorun yoksa butona tıklanıyor ve gönderi paylaşılıyor.

```
time.sleep(2)
post_send_button =
self.driver.find_element(By.CSS_SELECTOR, '.share-actions__primary-action
.artdeco-button.artdeco-button--2.artdeco-button--primary.ember-view')
# Gönderi paylaşma butonu görünür mü ve tıklanılabilir mi kontrol et
self.assertTrue(post_send_button.is_displayed())
self.assertTrue(post_send_button.is_enabled())
post_send_button.click();
```

Gönderi paylaş butonuna basıldıktan sonra modal kapanıyor. Daha sonra gönderinin paylaşılıp paylaşılmadığı, ilgili gönderinin içerisindeki yazı kontrol edilerek sağlanıyor.

```
# post paylaşımını kontrol etme
time.sleep(2)
post_content_div =
self.driver.find_element(By.CSS_SELECTOR, '.break-words>span')
self.assertTrue(post_content_div.is_displayed())
self.assertEqual(self.postText, post_content_div.text)
time.sleep(3)
```

Test sonucu ekran görüntüsü:

```
(base) furkangülse@furkanMAC python-testing % python linkedIn_test.py
.
-----
Ran 1 test in 44.116s
```

Amazonda Bulunan İlk Ürün ile Aranan Kitap Karşılaştırma Testi

Kullanılacak tarayıcı için gerekli dosya yolu tanımlanıyor ve açılıyor, sonrasında Url'si tanımlanan web sitesine gidiliyor ve tarayıcı tam ekran yapılıyor.

```
# Firefox tarayıcısı için gerekli geckodriver aranıyor
cls.driver =
webdriver.Firefox(executable_path="/Users/Yunus/Desktop/Test
dersi/geckodriver")
cls.driver.implicitly_wait(30)
# Amazon sayfasının url'si tanımlanıyor
cls.driver.get("https://www.amazon.com.tr/")
# Tarayıcı tam ekran olacak şekilde büyütülüyor
cls.driver.maximize_window()
```

Selenium, amazon.com.tr adresine giriş yaparak sayfanın doğru görüntülenip görüntülenmediğini kontrol ediyor ve verilen değeri input alanı içerisine yazılıyor.

```
first_book_title = "Şeker Portakalı"

input_alan =
```

```
self.driver.find_element(By.XPATH, "//input[@id='twotabsearchtextbox']")
# Arama input alanı erişilebilir mi kontrol et
self.assertTrue(input_alan.is_enabled())
# Arama kutusuna tıkla ve 'Kitap İsmi' gir
self.driver.find_element(By.XPATH, "//input[@id='twotabsearchtextbox']").
send_keys(first_book_title)
# Aramayı başlat
self.driver.find_element(By.XPATH, "//input[@id='twotabsearchtextbox']").
submit()
```

Ürün listeleme sayfasının yüklenmesi için 3 saniye bekleniliyor ve açıldıktan sonra sayfa üzerindeki ilk ürüne tıklanıyor.

```
# Çıkan ilk ürüne tıklanıyor
time.sleep(3)
self.driver.find_element(By.XPATH, "(//div[@class='sg-col-inner']//img[co
ntains(@data-image-latency, 's-product-image'))][1]").click()
```

Girilen ilk ürün ile aradığımız kitabın başlığı, yazarı, dili ve sayfa sayısı aynı mı teker teker kontrol ediliyor. Her kontrol sonunda 3 saniye bekleniyor ve işlem onay bilgisi yazdırılıyor.

```
# Çıkan ilk ürüne tıklıyorum
time.sleep(3)
self.driver.find_element(By.XPATH, "(//div[@class='sg-col-inner']//img[co
ntains(@data-image-latency, 's-product-image'))][1]").click()

# Ürünün başlığı kontrol
time.sleep(3)
title =
self.driver.find_element(By.XPATH, '//*[@id="productTitle"]').text.strip(
)
self.assertEqual(first_book_title, title)
print("Title done")

# Ürünün yazarı kontrol
time.sleep(3)
author =
self.driver.find_element(By.XPATH, '//*[@id="bylineInfo"]/span[1]/a').tex
t.strip()
self.assertEqual(author, "Jose Mauro De Vasconcelos")
print("Author done")
```

```
# Ürünün dili kontrol
time.sleep(3)
language = self.driver.find_element(By.XPATH,
'//*[@id="detailBullets_feature_div"]/ul/li[2]/span/span[2]').text.strip()
self.assertEqual(language, "Türkçe")
print("Language done")

# Ürünün sayfa sayısı kontrol
time.sleep(3)
pages =
self.driver.find_element(By.XPATH, '//*[@id="detailBullets_feature_div"]/
ul/li[3]/span/span[2]').text.strip()
pages = pages[:3]
self.assertEqual(pages, "184")
print("Pages done")
```

Web siteden çekilen veriler başlıkları ile yazdırılıyor.

```
print("Pages = " , pages)
print("Title = ", title)
print("Author = ", author)
print("Language = ", language)
```

Tüm işlemle bittiğinde tarayıcı kapatılıyor.

```
# Tarayıcı Kapatılıyor
cls.driver.quit()
```

Test sonucu ekran görüntüsü :

```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

===== RESTART: C:\Users\Yunus\Desktop\Test dersi\amazon_kitap.py =====
done
Title done
Author done
Language done
Pages done
Pages = 184
Title = Şeker Portakalı
Author = Jose Mauro De Vasconcelos
Language = Türkçe
.
-----
Ran 1 test in 32.899s

OK
|
```

Çiçek Sepeti Test

Çiçek Sepeti'nde 4 farklı test gerçekleştirdim.

Bunlar;

- Sisteme giriş yapması
- Sepete ürün eklemesi
- Sepete eklenen aynı iki ürünün fiyatını kontrol etme

Selenium ile Chrome directory oluşturarak Çiçek Sepeti sayfasına giriş sayfasının görüntülenmesi kontrol edilir.

```
s = Service('directory/chromedriver.exe')
browser = webdriver.Chrome(service=s)
browser.maximize_window()
url = 'https://www.ciceksepeti.com'
browser.get(url)

def sign(self):
    self.browser.get("https://www.ciceksepeti.com/uye-girisi")
    time.sleep(1)
```

Email input alanı ve şifre input alanı görünürlüğü test edilir, email ve şifre girilmesi gereken input alanlarına yazılır ve giriş yapılır.

```
userName = self.browser.find_element_by_name("Email")
userName.send_keys(email)
#email input alanı kontrol
```

```
emailInput = self.browser.find_element(By.ID, "EmailLogin")
self.assertTrue(emailInput.is_enabled())
time.sleep(2)
userPassword = self.browser.find_element_by_id("Password")
#password input alanı kontrol
passwordInput = self.browser.find_element(By.ID, "Password")
self.assertTrue(passwordInput.is_enabled())

userPassword.send_keys(password)
time.sleep(2)
userPassword.send_keys(Keys.ENTER)
time.sleep(1)
```

Giriş başarılı olarak gerçekleştikten sonra anasayfanın görünürlüğü kontrol edilir.

```
main_page = self.driver.browser.find_element(By.CLASS_NAME, "header__top")
self.assertTrue(main_page.is_displayed())
```

Arama alanında bulunan input alanına ürün adı girilerek ara butonuna tıklanır.

```
# aranacak ürün
product = "pembe lilyum çiçek buketi"

def searchProduct(self):
    search = self.browser.find_element(By.CLASS_NAME, "product-search__input")
    search.send_keys(product)
    self.browser.find_element(By.CLASS_NAME, "icon-search-444").click()
    time.sleep(2)
```

Açılan sayfada çıkan ürün aranan ürün ile adı aynı mı kontrol edilir ve ürün detay sayfası açılır.

```
self.browser.find_element(By.XPATH,
    "/html/body/main/div/div[4]/div[4]/div/div/a/div[1]/img").click()
name = self.browser.find_element(By.XPATH,
    "//*[ @id='productDetailSend']/div/div/div[2]/div[2]/div[2]/div/div[1]/h1/
    span").text.strip()
self.assertEqual("Pembe Lilyum Çiçek Buketi", name)
print("Product name success")
time.sleep(2)
```

Açılan detay sayfada teslimat tarih seçim butonları çalışıyor mu kontrol ediliyor ve teslimat tarih detayları seçimi yapıyor.

```
self.browser.find_element(By.CLASS_NAME, "js-emoji-characters").click()
time.sleep(1)
self.browser.find_element(By.CLASS_NAME,
"js-region-self-service").click()
time.sleep(2)
self.browser.find_element(By.XPATH,
"//*[@id='productDetailSend']/div/div/div[2]/div[2]/div[4]/div[5]/div[2]
/div[1]/span[1]").click()
time.sleep(2)
self.browser.find_element(By.XPATH,
"//*[@id='productDetailSend']/div/div/div[2]/div[2]/div[4]/div[13]/div/d
iv/div/div[1]/input").click()
time.sleep(2)
self.browser.find_element(By.XPATH,
"//*[@id='productDetailSend']/div/div/div[2]/div[2]/div[4]/div[13]/div/d
iv/div/div[2]/div/div[1]/p").click()
time.sleep(2)
```

Sepete ekleme butonu görünürlüğü kontrol ediliyor ve sipariş ver butonuna tıklanıyor.

```
#buton görünürlülüğü ve işlevselliği kontrol edilir
self.assertTrue(button.is_displayed())
self.assertTrue(button.is_enabled())
buy_product = self.browser.find_element(By.CLASS_NAME,
"js-add-to-cart").click()
```

Aynı üründen farklı tarihlerde teslimat için iki tane sepete eklenebilir mi kontrol edilir.

```
self.browser.back()
self.browser.find_element(By.CLASS_NAME, "js-emoji-characters").click()
time.sleep(1)
self.browser.find_element(By.CLASS_NAME,
"js-region-self-service").click()
time.sleep(2)
productPrice = self.browser.find_element(By.CLASS_NAME,
"js-price-integer").text
print("price: ", productPrice)
```

Ürünü sepete iki defa eklendiğinde sepette bulunan fiyatın doğruluğu kontrol edilir.

```
#ürün fiyatı string den int türüne çevrilir
intPrice = int(productPrice)
```



```

#sepete eklenecek aynı iki ürünün toplam fiyatı
tekliurunikiuruncarpimi = intPrice*2

print("sepete eklenecek aynı iki ürünün toplam fiyatı",
tekliurunikiuruncarpimi)

#sepete eklencek iki ürünün toplam fiyatının string e çevrilir
stringtekliurunikiuruncarpimi = str(tekliurunikiuruncarpimi)

self.browser.find_element(By.XPATH,
"//*[@id='productDetailSend']/div/div/div[2]/div[2]/div[4]/div[5]/div[2]/div[1]/span[1]").click()
time.sleep(2)
self.browser.find_element(By.XPATH,
"//*[@id='productDetailSend']/div/div/div[2]/div[2]/div[4]/div[13]/div/div/div/div[1]/input").click()
time.sleep(2)
self.browser.find_element(By.XPATH,
"//*[@id='productDetailSend']/div/div/div[2]/div[2]/div[4]/div[13]/div/div/div/div[2]/div/div[1]/p").click()
time.sleep(2)
self.browser.find_element(By.CLASS_NAME, "js-add-to-cart").click()
time.sleep(3)
self.browser.find_element(By.CLASS_NAME, "agree-button").click()
time.sleep(2)

#sepetteki toplam fiyat
totalPrice = self.browser.find_element(By.CLASS_NAME,
"grand-total-integer").text
print("total price", totalPrice)

#sepetteki toplam fiyatın string e çevrilir
strtotalPrice = str(totalPrice)
time.sleep(2)
self.assertEqual(stringtekliurunikiuruncarpimi, strtotalPrice)
print("price succeeded"

```

Tarayıcı kapatılır.

```
self.browser.close()
```

Test Sonucu Ekran Görüntüsü:

```
CicekSepeti  searchProduct()
Run: main x
C:\Users\USER\Desktop\CicekSepetiTest\venv\Scripts\python.exe C:/Users/USER/Desktop/CicekSepetiTest/main.py
Product name success
price: 139
sepete eklenecek aynı iki ürünün toplam fiyatı 278
total price 278
price succeeded

Process finished with exit code 0
```