**INTERNSHIP**

PRODIGY\_ST\_05\_Automated UI Tests using any testing tool for an E-Commerce Checkout Flow

**CODE:**

import time

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

# --- Setup Chrome Options to Disable Password Manager ---

chrome\_options = webdriver.ChromeOptions()

chrome\_options.add\_experimental\_option("prefs", {

"credentials\_enable\_service": False,

"profile.password\_manager\_enabled": False

})

# Initialize the Chrome WebDriver

driver = webdriver.Chrome(options=chrome\_options)

driver.maximize\_window()

# Define an explicit wait for robust element location

wait = WebDriverWait(driver, 20)

# Define website URL and user credentials

BASE\_URL = "https://www.saucedemo.com/"

USERNAME = "standard\_user"

PASSWORD = "secret\_sauce"

def run\_checkout\_test():

try:

print("Step 1: Navigating to the website and logging in...")

driver.get(BASE\_URL)

wait.until(EC.visibility\_of\_element\_located((By.ID, "user-name"))).send\_keys(USERNAME)

driver.find\_element(By.ID, "password").send\_keys(PASSWORD)

driver.find\_element(By.ID, "login-button").click()

wait.until(EC.url\_contains("inventory.html"))

print("✅ Verification PASSED: Successfully logged in and redirected to inventory page.")

print("\nStep 2: Adding an item to the cart...")

driver.find\_element(By.ID, "add-to-cart-sauce-labs-backpack").click()

shopping\_cart\_badge = driver.find\_element(By.CLASS\_NAME, "shopping\_cart\_badge")

assert shopping\_cart\_badge.text == "1"

print("✅ Verification PASSED: Item count in cart is 1.")

shopping\_cart\_badge.click()

wait.until(EC.url\_contains("cart.html"))

item\_in\_cart = wait.until(EC.visibility\_of\_element\_located((By.CLASS\_NAME, "inventory\_item\_name")))

assert "Sauce Labs Backpack" in item\_in\_cart.text

print("✅ Verification PASSED: Successfully navigated to the cart page with the correct item.")

print("\nStep 3: Proceeding to checkout...")

driver.find\_element(By.ID, "checkout").click()

wait.until(EC.visibility\_of\_element\_located((By.ID, "first-name")))

assert "checkout-step-one.html" in driver.current\_url

print("✅ Verification PASSED: Navigated to the checkout information page.")

print("\nStep 4: Filling form and verifying checkout overview...")

driver.find\_element(By.ID, "first-name").send\_keys("John")

driver.find\_element(By.ID, "last-name").send\_keys("Doe")

driver.find\_element(By.ID, "postal-code").send\_keys("12345")

driver.find\_element(By.ID, "continue").click()

wait.until(EC.visibility\_of\_element\_located((By.ID, "finish")))

assert "checkout-step-two.html" in driver.current\_url

print("✅ Verification PASSED: Checkout overview page loaded successfully.")

print("\nStep 5: Finalizing the purchase...")

finish\_button = driver.find\_element(By.ID, "finish")

driver.execute\_script("arguments[0].click();", finish\_button)

success\_header = wait.until(EC.visibility\_of\_element\_located((By.CLASS\_NAME, "complete-header")))

assert "checkout-complete.html" in driver.current\_url

assert "THANK YOU FOR YOUR ORDER" in success\_header.text.upper()

print("✅ Verification PASSED: Order confirmation and success message verified.")

print("\n🎉 AUTOMATION TEST COMPLETED SUCCESSFULLY! 🎉")

except Exception as e:

timestamp = time.strftime("%Y%m%d-%H%M%S")

screenshot\_name = f"failure\_{timestamp}.png"

driver.save\_screenshot(screenshot\_name)

print(f"\n❌ TEST FAILED: An error occurred.")

print(f"📄 Screenshot saved as {screenshot\_name}")

print(f"Error details: {e}")

finally:

print("\nClosing the browser.")

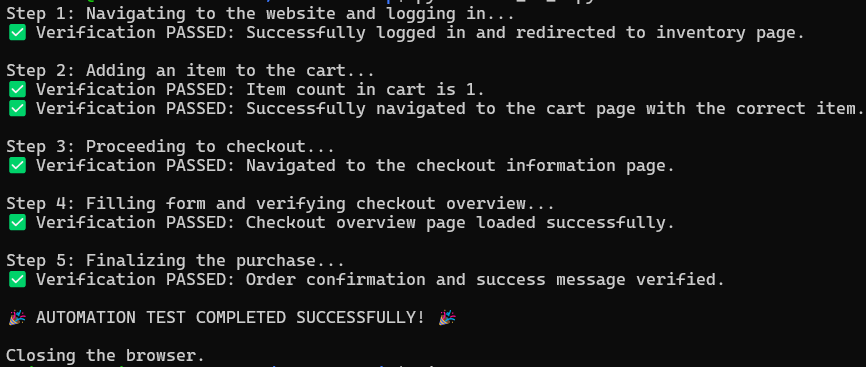
driver.quit()

# --- Run the test ---

if \_\_name\_\_ == "\_\_main\_\_":

run\_checkout\_test()

**OUTPUT:**

****

Aritra Datta