EX.NO.: 05
DATE: 16.07.2025

SELENIUM FOR E-COMMERCE PLATFORM

AIM

To automate the following end-to-end test on the Automation Exercise Website using Selenium WebDriver:

1. Login with valid credentials.
2. Navigate to the Products page.
3. Add a product to the cart.
4. Proceed to checkout and simulate a payment.
5. Download the invoice.
6. Logout from the application.

ALGORITHM

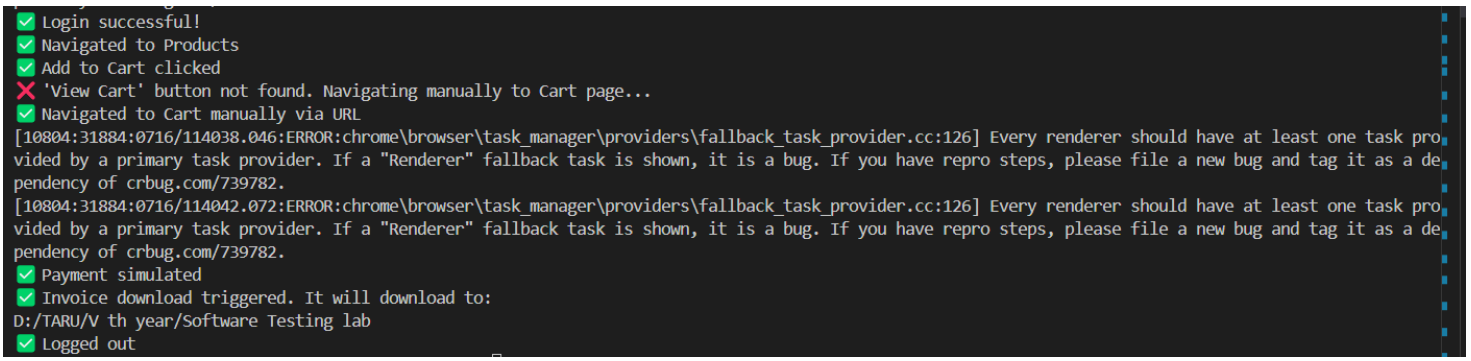
1. Start browser session using Selenium WebDriver.
2. Navigate to the home page of `https://automationexercise.com`.
3. Login using provided credentials.
4. Navigate to Products via link or XPath fallback.
5. Add the first product on the page to the cart.
6. View the cart and validate the item is present.
7. Proceed to checkout, fill out order message, and place order.
8. Simulate payment with dummy card details.
9. Download invoice.
10. Logout and close the browser.

CODE AND OUTPUT

```
from selenium import webdriver
from selenium.webdriver.edge.service import Service as EdgeService
from selenium.webdriver.edge.options import Options
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import time

download_dir = r"D:/TARU/V th year/Software Testing lab"

options = Options()
options.add_experimental_option("prefs", {
    "download.default_directory": download_dir,
    "download.prompt_for_download": False,
    "download.directory_upgrade": True,
    "safebrowsing.enabled": True
})

edge_driver_path = r"D:/TARU/V th year/Software Testing
lab/edgedriver_win64/msedgedriver.exe"
service = EdgeService(edge_driver_path)
driver = webdriver.Edge(service=service, options=options)

driver.maximize_window()
```

```
driver.get("https://automationexercise.com")
wait = WebDriverWait(driver, 10)

# --- LOGIN ---
driver.find_element(By.LINK_TEXT, "Signup / Login").click()
wait.until(EC.presence_of_element_located((By.NAME, "email")))
driver.find_element(By.NAME, "email").send_keys("tarunika.test@gmail.com")
driver.find_element(By.NAME, "password").send_keys("tarunika.test")
driver.find_element(By.XPATH, "//button[text()='Login']").click()

try:
    wait.until(EC.presence_of_element_located((By.LINK_TEXT, "Logout")))
    print("✅ Login successful!")
except:
    print("❌ Login failed. Please check credentials or site.")
    driver.quit()
    exit()

# --- ADD TO CART ---
try:
    products_link = wait.until(EC.element_to_be_clickable((By.PARTIAL_LINK_TEXT,
"Product")))
    products_link.click()
    print("✅ Navigated to Products")
except:
    print("❌ 'Products' link not found. Trying XPath fallback...")
    try:
        fallback = wait.until(EC.element_to_be_clickable((By.XPATH,
"//a[@href='/products']")))
        fallback.click()
        print("✅ Navigated to Products using fallback XPath")
    except:
        print("❌ Failed to locate Products link. Exiting.")
        driver.quit()
        exit()

time.sleep(2)
driver.execute_script("window.scrollTo(0, 600)")
driver.find_element(By.XPATH, "(//a[@class='btn btn-default add-to-cart'])[1]").click()
print("✅ Add to Cart clicked")

# --- VIEW CART ---
try:
    time.sleep(1)
    view_cart = WebDriverWait(driver, 5).until(
        EC.element_to_be_clickable((By.XPATH, "//a[contains(text(), 'View Cart')]"))
    )
    view_cart.click()
    print("✅ Product added to cart and navigated to cart page")
```

```
except:
    print("❌ 'View Cart' button not found. Navigating manually to Cart page...")
    try:
        driver.get("https://automationexercise.com/view_cart")
        wait.until(EC.presence_of_element_located((By.LINK_TEXT, "Proceed To
Checkout")))
        print("✅ Navigated to Cart manually via URL")
    except:
        print("❌ Failed to load Cart page manually. Exiting.")
        driver.quit()
        exit()

# --- CHECKOUT ---
try:
    driver.find_element(By.LINK_TEXT, "Proceed To Checkout").click()
    wait.until(EC.presence_of_element_located((By.NAME, "message")))
    driver.find_element(By.NAME, "message").send_keys("Please deliver soon.")
    driver.find_element(By.XPATH, "//a[text()='Place Order']").click()
except:
    print("❌ Checkout failed. Please check site layout.")
    driver.quit()
    exit()

# --- SIMULATE PAYMENT ---
driver.find_element(By.NAME, "name_on_card").send_keys("Test User")
driver.find_element(By.NAME, "card_number").send_keys("4111111111111111")
driver.find_element(By.NAME, "cvc").send_keys("123")
driver.find_element(By.NAME, "expiry_month").send_keys("12")
driver.find_element(By.NAME, "expiry_year").send_keys("2026")
driver.find_element(By.ID, "submit").click()
print("✅ Payment simulated")

# --- DOWNLOAD INVOICE ---
try:
    download_btn = wait.until(EC.element_to_be_clickable((By.LINK_TEXT, "Download
Invoice")))
    download_btn.click()
    print(f"✅ Invoice download triggered. It will download to:\n{download_dir}")
    time.sleep(2)
except:
    print("❌ Failed to download invoice")

# --- LOGOUT ---
driver.find_element(By.LINK_TEXT, "Logout").click()
print("✅ Logged out")

driver.quit()
```



INFERENCE

The automation script successfully simulates a complete user journey from login to payment on the Automation Exercise site. All critical steps (login, add to cart, checkout, payment, invoice download) passed without manual intervention. The test validates that the user flow and site functionalities are working as expected.