# PRODIGY – SOFTWARE TESTING INTERNSHIP

**TASK 5 – Automated UI Tests for E-Commerce Checkout Flow**

**Application URL:** http://www.automationpractice.pl/index.php

**Testing Tool : Selenium WebDriver with Python**

Done By : Intern Lalitha B, Prodigy

(01/01/2026 – 31/01/2026)

## 1. Introduction :

E-commerce applications involve multiple user interactions such as browsing products, adding items to a cart, filling checkout forms, and completing payments. Any failure in these steps can lead to loss of customers.

This task focuses on **automating the complete checkout process** of an e-commerce website using Selenium WebDriver to ensure that all critical functionalities work as expected.

## 2. Objective of Automation :

The objectives of this automation task are:

- To automate the end-to-end checkout flow
- To verify each page transition during checkout
- To validate mandatory checkout form fields
- To ensure products can be added to the cart successfully
- To verify successful order placement and confirmation message

## 3. Application Under Test (AUT) :

| Parameter | Details |
|---|---|
| Application Name | Automation Practice |
| URL | http://www.automationpractice.pl/index.php |
| Type | Web-based E-Commerce Application |
| Features Tested | Login, Cart, Checkout, Payment |

## 4.Test Environment :

| Parameter | Details |
|---|---|
| Operating System | Windows / macOS |
| Browser | Google Chrome |
| Automation Tool | Selenium WebDriver |
| Programming Language | Python |
| IDE | Visual Studio Code |
| Testing Type | Automated UI Testing |

## 5.Test Data Used :

| Field | Value |
|---|---|
| Email | testuser123@gmail.com |
| Password | Test@123 |
| Payment | Bank Wire |

## 6. Scope of Testing :

In-Scope

- User login
- Product selection
- Add to cart
- Checkout process
- Address and shipping confirmation
- Payment processing
- Order success verification

Out-of-Scope

- Performance testing
- Security testing
- Mobile responsiveness
- Backend validation

## 7. Automation Script Implementation :

**Key Functionalities Automated:**

- Login to the application
- Add product to cart
- Navigate through checkout pages
- Fill and validate checkout forms
- Complete payment
- Verify success message

**Script Used:**

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
import time
import os

if not os.path.exists("screenshots"):
    os.makedirs("screenshots")

driver = webdriver.Chrome()
driver.maximize_window()

driver.get("http://www.automationpractice.pl/index.php")
time.sleep(2)

# Login
driver.find_element(By.CLASS_NAME, "login").click()
driver.find_element(By.ID, "email").send_keys("testuser123@gmail.com")
driver.find_element(By.ID, "passwd").send_keys("Test@123")
```

```
driver.find_element(By.ID, "SubmitLogin").click()
time.sleep(3)
driver.save_screenshot("screenshots/login.png")

# Add product to cart
product = driver.find_element(By.CSS_SELECTOR, ".product-container")
ActionChains(driver).move_to_element(product).perform()
driver.find_element(By.CSS_SELECTOR, ".ajax_add_to_cart_button").click()
time.sleep(3)
driver.save_screenshot("screenshots/product_added.png")

# Checkout flow
driver.find_element(By.XPATH, "//a[@title='Proceed to checkout']").click()
time.sleep(2)
driver.find_element(By.XPATH, "//a[contains(@class,'standard-checkout')]").click()
time.sleep(2)
driver.save_screenshot("screenshots/address.png")

driver.find_element(By.NAME, "processAddress").click()
time.sleep(2)

driver.find_element(By.ID, "cgv").click()
driver.save_screenshot("screenshots/shipping.png")
driver.find_element(By.NAME, "processCarrier").click()
time.sleep(2)

driver.find_element(By.CLASS_NAME, "bankwire").click()
driver.save_screenshot("screenshots/payment.png")
```

```
driver.find_element(By.XPATH, "//button[contains(@class,'button-medium')]").click()
time.sleep(3)

success_msg = driver.find_element(By.CSS_SELECTOR, ".cheque-indent strong").text
driver.save_screenshot("screenshots/order_success.png")

if "complete" in success_msg.lower():
    print("Order placed successfully")

driver.quit()
```

## 8. Verification Of Page Transmission :

| Page | Verification |
|---|---|
| Login Page | User logged in successfully |
| Cart Summary | Product displayed correctly |
| Address Page | Address details loaded |
| Shipping Page | Terms checkbox validated |
| Payment Page | Payment option selected |
| Confirmation Page | Order success message shown |

## 10. Form Validation Checks :

- Mandatory login fields validated
- Checkout cannot proceed without accepting Terms & Conditions
- Payment selection required before confirmation
- Success message displayed only after completing all steps

## 11. Issues Encountered During Automation :

| Issue | Description | Resolution |
|---|---|---|
| Dynamic elements | Product hover required | Used ActionChains |
| Slow page loading | Checkout delay | Used time.sleep |
| Mandatory checkbox | Shipping page blocked | Added checkbox validation |

## 12. Test Execution Evidence :

Screenshots were captured at each major step:

- Login confirmation
- Product added to cart
- Address confirmation
- Shipping validation
- Payment selection
- Order success message

These screenshots serve as **proof of successful automation execution**.

## 13. Test Execution Result :

| Test Scenario | Status |
|---|---|
| Login | Pass |
| Add to Cart | Pass |
| Checkout Pages | Pass |
| Form Validation | Pass |
| Order Confirmation | Pass |

## 14. Conclusion :

The automation script successfully validated the complete checkout workflow of the e-commerce application.
All page transitions, form validations, and order confirmation steps were executed and verified successfully with proper screenshot evidence.

This task demonstrates effective use of Selenium WebDriver for real-world UI automation testing.

## 15. Future Enhancements :

- Integrate PyTest framework
- Generate HTML test reports
- Add negative checkout test cases
- Implement data-driven testing