TASK 5 :

**Automated UI** **Tests using any** **testing tool for**  
**an E-Commerce** **Checkout Flow**

**Automate the checkout process on any e-commerce website using any tool of your choice. Your script should include adding items to the cart, filling out checkout forms, and verifying the success message after purchase.**

**Requirements:**

* **Verify each page transition and form** **validation message.**
* **Document your script and output, noting any issues encountered during automation.**

import pytest

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

# Replace with the actual URL of the e-commerce website

BASE\_URL = "https://www.example-ecommerce.com"

# Replace with the actual locators for elements on the website

PRODUCT\_ITEM = "//div[@class='product-item']"

ADD\_TO\_CART\_BUTTON = "//button[contains(text(), 'Add to Cart')]"

CART\_LINK = "//a[contains(text(), 'Cart')]"

CHECKOUT\_BUTTON = "//button[contains(text(), 'Checkout')]"

FIRST\_NAME\_FIELD = "firstName"

LAST\_NAME\_FIELD = "lastName"

EMAIL\_FIELD = "email"

ADDRESS\_FIELD = "address"

CITY\_FIELD = "city"

STATE\_FIELD = "state"

ZIP\_CODE\_FIELD = "zipCode"

CARD\_NUMBER\_FIELD = "cardNumber"

EXPIRY\_DATE\_FIELD = "expiryDate"

CVV\_FIELD = "cvv"

PLACE\_ORDER\_BUTTON = "//button[contains(text(), 'Place Order')]"

SUCCESS\_MESSAGE = "Your order has been placed successfully!"

@pytest.fixture(scope="class")

def driver():

driver = webdriver.Chrome() # Replace with your preferred driver

yield driver

driver.quit()

def test\_checkout\_process(driver):

driver.get(BASE\_URL)

# Add an item to the cart

product\_item = driver.find\_element(By.XPATH, PRODUCT\_ITEM)

product\_item.find\_element(By.XPATH, ADD\_TO\_CART\_BUTTON).click()

# Navigate to the cart page

driver.find\_element(By.XPATH, CART\_LINK).click()

WebDriverWait(driver, 10).until(EC.presence\_of\_element\_located((By.XPATH, CHECKOUT\_BUTTON)))

# Proceed to checkout

driver.find\_element(By.XPATH, CHECKOUT\_BUTTON).click()

# Fill out checkout form

driver.find\_element(By.ID, FIRST\_NAME\_FIELD).send\_keys("John")

driver.find\_element(By.ID, LAST\_NAME\_FIELD).send\_keys("Doe")

driver.find\_element(By.ID, EMAIL\_FIELD).send\_keys("john.doe@example.com")

driver.find\_element(By.ID, ADDRESS\_FIELD).send\_keys("123 Main St")

driver.find\_element(By.ID, CITY\_FIELD).send\_keys("Anytown")

driver.find\_element(By.ID, STATE\_FIELD).send\_keys("CA")

driver.find\_element(By.ID, ZIP\_CODE\_FIELD).send\_keys("12345")

driver.find\_element(By.ID, CARD\_NUMBER\_FIELD).send\_keys("1234567890123456")

driver.find\_element(By.ID, EXPIRY\_DATE\_FIELD).send\_keys("12/25")

driver.find\_element(By.ID, CVV\_FIELD).send\_keys("123")

# Place the order

driver.find\_element(By.XPATH, PLACE\_ORDER\_BUTTON).click()

# Verify success message

WebDriverWait(driver, 10).until(EC.text\_to\_be\_present\_in\_element((By.XPATH, "//div[@class='success-message']"), SUCCESS\_MESSAGE))

assert SUCCESS\_MESSAGE in driver.page\_source

# Verify page transitions (example - check for cart page after adding item)

assert CART\_LINK in driver.page\_source

# Verify form validation messages (example - empty first name field)

driver.find\_element(By.ID, FIRST\_NAME\_FIELD).clear()

driver.find\_element(By.XPATH, PLACE\_ORDER\_BUTTON).click()

assert "Please enter your first name" in driver.page\_source