

# Part 1: Web Application Automation (Selenium)

## Task 1: Automate a Login Flow

```
from selenium import webdriver

from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time

driver = webdriver.Chrome()

driver.get("https://example-ecommerce-site.com/login")

username = driver.find_element(By.ID, "username")
password = driver.find_element(By.ID, "password")

username.send_keys("testuser")
password.send_keys("password123")

login_button = driver.find_element(By.ID, "login-button")
login_button.click()

time.sleep(2)

try:
    welcome_message = driver.find_element(By.XPATH, "//h1[contains(text(),'Welcome')]")
    assert "Welcome" in welcome_message.text
    print("Login successful!")
except AssertionError:
    print("Login failed.")

driver.quit()
```

## Task 2: Product Search Automation

```
from selenium import webdriver

from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time

driver = webdriver.Chrome()

driver.get("https://example-ecommerce-site.com/login")
search_box = driver.find_element(By.ID, "search-bar")
search_box.send_keys("laptop")
search_box.send_keys(Keys.RETURN)
time.sleep(2)

search_results = driver.find_elements(By.CSS_SELECTOR, ".product-item")

if len(search_results) > 0:
    print("Search results found.")
else:
    print("No search results found.")
    driver.quit()

for index, product in enumerate(search_results[:3]):
    product_name = product.find_element(By.CSS_SELECTOR, ".product-name").text
    product_price = product.find_element(By.CSS_SELECTOR, ".product-price").text
    print(f"Product {index + 1}: Name - {product_name}, Price - {product_price}")
driver.quit()
```