Automated Test Cases:

Test 1: User Login with Valid Credentials

- **Scenario:** Login with a registered user.
- **Why Automate:** Frequent regression check — ensures authentication works.

```
import pytest
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.alert import Alert
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC


# ------------------------
# Pytest Fixture: WebDriver Setup & Teardown
# ------------------------
@pytest.fixture
def driver():
    driver = webdriver.Chrome()
    driver.maximize_window()
    driver.get("https://www.demoblaze.com/")
    yield driver
    driver.quit()


# ------------------------
# Helper: Wait for element by locator
# ------------------------
def wait_for_element(driver, locator, timeout=10):
    return WebDriverWait(driver, timeout).until(EC.presence_of_element_located(locator))


# ------------------------
# Test 1: Valid Login
# ------------------------
def test_login_valid_user(driver):
    wait_for_element(driver, (By.ID, "login2")).click()
    wait_for_element(driver, (By.ID, "loginusername")).send_keys("testuser")  # Replace with
valid user
    driver.find_element(By.ID, "loginpassword").send_keys("testpass")  # Replace with valid
password
```

```
    driver.find_element(By.XPATH, "//button[text()='Log in']").click()

    # Wait until welcome message appears
    WebDriverWait(driver, 10).until(EC.text_to_be_present_in_element((By.ID, "nameofuser"),
"Welcome"))
    assert "Welcome testuser" in driver.page_source
```

Test 2: Invalid Login Attempt

- **Scenario:** Try to log in with wrong credentials.
- **Why Automate:** Security & negative path validation.

```
# ------------------------
# Test 2: Invalid Login Attempt
# ------------------------
def test_login_invalid_user(driver):
    wait_for_element(driver, (By.ID, "login2")).click()
    wait_for_element(driver, (By.ID, "loginusername")).send_keys("wronguser")
    driver.find_element(By.ID, "loginpassword").send_keys("wrongpass")
    driver.find_element(By.XPATH, "//button[text()='Log in']").click()

    # Wait for alert and check its text
    WebDriverWait(driver, 10).until(EC.alert_is_present())
    alert = Alert(driver)
    assert "User does not exist" in alert.text
    alert.accept()
```

Test 3: Add Product to Cart

- **Scenario:** Add a product from "Laptops" category to the cart and verify it appears in the cart.
- **Why Automate:** Core business flow — high priority regression candidate.

```
# ------------------------
# Test 3: Add Product to Cart
# ------------------------
def test_add_product_to_cart(driver):
    wait_for_element(driver, (By.LINK_TEXT, "Laptops")).click()
    wait_for_element(driver, (By.LINK_TEXT, "Sony vaio i5")).click()
    wait_for_element(driver, (By.LINK_TEXT, "Add to cart")).click()
    # Handle alert after adding to cart
    WebDriverWait(driver, 10).until(EC.alert_is_present())
    Alert(driver).accept()

    wait_for_element(driver, (By.ID, "cartur")).click()
    assert "Sony vaio i5" in driver.page_source
```

Test 4: Remove Product from Cart

- **Scenario:** Add two products, remove one, and verify cart is updated.
- **Why Automate:** Validates cart consistency.

```
# -----------------------
# Test 4: Remove Product from Cart
# -----------------------
def test_remove_product_from_cart(driver):
    # Step 1: Add product
    wait_for_element(driver, (By.LINK_TEXT, "Laptops")).click()
    wait_for_element(driver, (By.LINK_TEXT, "Sony vaio i5")).click()
    wait_for_element(driver, (By.LINK_TEXT, "Add to cart")).click()
    WebDriverWait(driver, 10).until(EC.alert_is_present())
    Alert(driver).accept()

    wait_for_element(driver, (By.ID, "cartur")).click()

    # Step 2: Delete product from cart
    wait_for_element(driver, (By.XPATH, "//a[text()='Delete']")).click()

    # Wait until cart refreshes (element gone)
    WebDriverWait(driver, 10).until_not(EC.text_to_be_present_in_element((By.ID, "tbodyid"), "Sony vaio i5"))
    assert "Sony vaio i5" not in driver.page_source
```

Test 5: Place an Order (Checkout Flow)

- **Scenario:** Add product → go to cart → place order → confirm with details.
- **Why Automate**: End-to-end functional coverage of the main purchase journey.

```
# -----------------------
# Test 5: Place an Order (Checkout Flow)
# -----------------------
def test_place_order(driver):
    # Step 1: Add product
    wait_for_element(driver, (By.LINK_TEXT, "Phones")).click()
    wait_for_element(driver, (By.LINK_TEXT, "Samsung galaxy s6")).click()
    wait_for_element(driver, (By.LINK_TEXT, "Add to cart")).click()
    WebDriverWait(driver, 10).until(EC.alert_is_present())
    Alert(driver).accept()

    # Step 2: Go to Cart
    wait_for_element(driver, (By.ID, "cartur")).click()

    # Step 3: Place Order
    wait_for_element(driver, (By.XPATH, "//button[text()='Place Order']")).click()

    # Step 4: Fill order form
    wait_for_element(driver, (By.ID, "name")).send_keys("John Doe")
    driver.find_element(By.ID, "country").send_keys("USA")
    driver.find_element(By.ID, "city").send_keys("New York")
    driver.find_element(By.ID, "card").send_keys("1234567812345678")
    driver.find_element(By.ID, "month").send_keys("12")
    driver.find_element(By.ID, "year").send_keys("2025")
    driver.find_element(By.XPATH, "//button[text()='Purchase']").click()

    # Step 5: Confirm success message
    WebDriverWait(driver, 10).until(EC.visibility_of_element_located((By.CLASS_NAME,
"sweet-alert")))
    assert "Thank you for your purchase!" in driver.page_source

    driver.find_element(By.XPATH, "//button[text()='OK']").click()
```