**课程设计**

题目： Demoblaze网站的测试用例设计

**标题：Demoblaze网站测试用例设计**

1. **项目背景**

Demoblaze是一个在线购物网站，提供手机、笔记本电脑和显示器等商品。该网站模拟了一个电商平台，包含注册、登录、商品浏览、添加到购物车、管理购物车删除商品、商品分类、订单结算等完整流程。

主要功能：用户注册和登录、商品浏览、商品分类、添加到购物车、订单结算。

运行环境：平台Pycharm、浏览器Edge、操作系统Windows。

1. **测试目标**

用户注册：验证用户注册功能的正确性与安全性。

用户登录：验证用户登录功能的正确性与安全性。

商品浏览：验证商品浏览功能的准确性。

添加到购物车：验证添加商品至购物车的正确性和流畅性。

管理购物车：验证购物车管理的正确性和流畅性。

点击分类页商品：验证点击分类页商品的正确性和流畅性。

提交订单：订单提交流程的正确性和流畅性。

识别并报告任何功能缺陷，确保用户能够正常使用网站各项核心功能。

1. **测试范围**

**模块：**

注册模块

登录模块

商品浏览模块

购物车模块

商品分类模块

订单结算模块

**功能点：**

用户注册

用户登录

浏览商品列表

查看商品详情

添加商品到购物车

删除购物车商品

点击分类页商品

填写订单信息

提交订单

1. **测试用例设计**
2. **黑盒测试**
3. **注册功能**

针对 Demoblaze 网站的注册功能，我们使用等价类划分和边界值分析方法设计测试用例。 注册功能允许用户创建新的账户，以便进行后续的购物操作。注册时，用户需要提供用户名和密码。因此，我们需要验证用户在输入有效和无效的用户名和密码时，系统是否能够正确处理，并给出相应的提示信息。

**等价类划分（注册功能）**

|  |  |  |
| --- | --- | --- |
| **输入条件** | **有效等价类** | **无效等价类** |
| 用户名 | 非空字符串 | 空输入 |
| 用户名 | 未被注册的用户名 | 已被注册的用户名 |
| 密码 | 非空字符串 | 空输入 |
| 密码 | 符合密码复杂度要求 (如长度、包含数字/字母) | 不符合密码复杂度要求 |
| 组合验证 | 有效用户名 + 有效密码 | 有效用户名 + 无效密码 等所有无效组合情况 |

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | **用例** | **预期结果** | **实际结果** |
| 01 | 用户名：未注册用户名 密码：符合要求的密码 | 注册成功，弹出提示框 | 成功 |
| 02 | 用户名：已注册用户名 密码：符合要求的密码 | 注册失败，弹出提示框 | 失败 |
| 03 | 用户名：未注册用户名 密码：不符合要求的密码 | 注册失败，弹出提示框 | 失败 |
| 04 | 用户名：空 密码：符合要求的密码 | 注册失败，弹出提示框 | 失败 |
| 05 | 用户名：符合要求的用户名 密码：空 | 注册失败，弹出提示框 | 失败 |
| 06 | 用户名：空 密码：空 | 注册失败，弹出提示框 | 失败 |

1. **登录功能**

针对 Demoblaze 网站的登录功能，我们使用等价类划分方法设计测试用例。登录功能允许用户使用已注册的账户访问网站的受保护区域。登录时，用户需要提供用户名和密码。 因此，我们需要验证用户在输入有效和无效的用户名和密码时，系统是否能够正确处理，并给出相应的提示信息。

**等价类划分（登录功能）**

|  |  |  |
| --- | --- | --- |
| **输入条件** | **有效等价类** | **无效等价类** |
| 用户名 | 非空字符串 | 空输入 |
| 用户名 | 已注册的用户名 | 未注册的用户名 |
| 密码 | 非空字符串 | 空输入 |
| 密码 | 与注册用户名对应的正确密码 | 与注册用户名不对应的错误密码 |
| 组合验证 | 有效用户名 + 有效密码 | 有效用户名 + 无效密码 等无效组合情况 |

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | **用例** | **预期结果** | **实际结果** |
| 01 | 用户名：已注册用户名 密码：与该用户名对应的正确密码 | 登录成功，页面显示用户名 | 成功 |
| 02 | 用户名：已注册用户名 密码：与该用户名不对应的错误密码 | 登录失败，弹出提示框 | 失败 |
| 03 | 用户名：未注册用户名 密码：任意 | 登录失败，弹出提示框 | 失败 |
| 04 | 用户名：空 密码：任意 | 登录失败，弹出提示框 | 失败 |
| 05 | 用户名：任意 密码：空 | 登录失败，弹出提示框 | 失败 |
| 06 | 用户名：空 密码：空 | 登录失败，弹出提示框 | 失败 |

1. **添加/删除 购物车商品数量**

针对 Demoblaze 网站的购物车功能，我们使用边界值分析方法设计测试用例。购物车功能允许用户添加和删除商品。 在添加商品时，我们需要验证商品数量的边界值，例如最小值和最大值。 在删除商品时，我们需要验证删除单个商品和删除所有商品的情况。

**边界值分析（添加/删除 购物车商品数量）**

|  |  |  |  |
| --- | --- | --- | --- |
| **输入条件** | **边界值** | **测试用例** | **预期结果** |
| 商品数量 | 最小值：1 | 添加1件商品 | 购物车显示 "1" |
|  | 最大值：N (假设) | 添加N件商品 | 购物车显示 "N" |
|  | 最大值+1：N+1 | 尝试添加第N+1件商品 | 提示“库存不足”或类似信息 |
| 删除商品 | 删除1件 | 从购物车删除1件商品 | 购物车数量相应减少 |
| 删除商品 | 全部删除 | 从购物车删除所有商品 | 购物车为空，提示"购物车空空如也" |

|  |  |  |
| --- | --- | --- |
| **用例编号** | **用例** | **预期结果** |
| 01 | 购物车内有3件商品，删除其中1件 | 购物车显示数量变为2 |
| 02 | 购物车内有1件商品，删除该商品 | 购物车为空，提示"购物车空空如也" |
| 03 | 尝试添加超过库存数量的商品到购物车 | 提示“库存不足”或类似信息 |

1. **订单总价计算**

针对 Demoblaze 网站的订单结算功能，我们使用等价类划分方法设计测试用例。订单结算功能允许用户查看购物车中的商品，并计算订单的总价。 因此，我们需要验证在输入有效和无效的商品价格和商品数量时，系统是否能够正确计算订单总价，并给出相应的提示信息。

**等价类划分（订单总价计算）**

|  |  |  |
| --- | --- | --- |
| **输入条件** | **有效等价类** | **无效等价类** |
| 单件商品价格 | 正确的商品价格（正数） | 负数，零，非数字 |
| 商品数量 | 正确的商品数量（整数） | 负数，零，非整数 |
| 组合验证 | 单件商品价格 + 商品数量 | 单件商品价格+非法字符,等所有无效组合情况 |

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | **用例** | **预期结果** | **实际结果** |
| 01 | 商品价格：500 商品数量：2 | 订单总价：1000 | 成功 |
| 02 | 商品价格：250 商品数量：3 | 订单总价：750 | 成功 |
| 03 | 商品价格：500 商品数量：-2 | 订单总价：总价为负不成立,提示用户 | 失败 |
| 04 | 商品价格：abc 商品数量：3 | 订单总价：商品价格格式错误提示用户 | 失败 |

1. **白盒测试**
   * 1. **注册功能**

**JavaScript 逻辑（register.js）：**

|  |
| --- |
| function register(username, password) {  if (username === "" || password === "") {  showError("Username and password are required.");  return false;  }  if (usernameExists(username)) {  showError("This username already exists.");  return false;  }  // ... 其他注册逻辑 ...  return true;  }  content\_copydownload  Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).JavaScript |

**条件组合覆盖：**

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | **用户名** | **密码** | **预期结果** |
| 01 | "" | "" | 注册失败，提示用户名和密码不能为空 |
| 02 | "test" | "" | 注册失败，提示密码不能为空 |
| 03 | "" | "test" | 注册失败，提示用户名不能为空 |
| 04 | "test" | "test" | 注册成功 |
| 05 | "已存在用户名" | "test" | 注册失败，提示用户名已存在 |

* + 1. **登录功能**

**JavaScript 逻辑（login.js）：**

|  |
| --- |
| function login(username, password) {  if (username === "" || password === "") {  showError("Username and password are required.");  return false;  }  if (!userExists(username)) {  showError("Invalid username or password.");  return false;  }  if (getPassword(username) !== password) {  showError("Invalid username or password.");  return false;  }  // ... 其他登录逻辑 ...  return true;  }  content\_copydownload  Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).JavaScript |

**条件组合覆盖：**

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | **用户名** | **密码** | **预期结果** |
| 01 | "" | "" | 登录失败，提示用户名和密码不能为空 |
| 02 | "test" | "" | 登录失败，提示密码不能为空 |
| 03 | "" | "test" | 登录失败，提示用户名不能为空 |
| 04 | "test" | "test" | 登录失败，提示用户名或密码错误 |
| 05 | "已存在用户名" | "密码" | 登录失败，提示用户名或密码错误 |
| 06 | "已存在用户名" | "正确密码" | 登录成功 |

* + 1. **语句覆盖（登录逻辑）**

**Python 代码:**

|  |
| --- |
| def login(username, password):  if username == "valid\_user": # 语句1  if password == "valid\_password": # 语句2  return "登录成功"  else:  return "密码错误"  else:  return "用户名错误"  content\_copydownload  Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).Python |

**测试用例:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | **用户名** | **密码** | **预期结果** | **实际结果** |
| 01 | "valid\_user" | "valid\_password" | "登录成功" | 成功 |
| 02 | "valid\_user" | "wrong\_pass" | "密码错误" | 失败 |
| 03 | "invalid\_user" | "valid\_password" | "用户名错误" | 失败 |

* + 1. **判定覆盖（商品分类）**

Demoblaze 网站有以下商品分类：

Phones

Laptops

Monitors

**JavaScript 逻辑（category.js）：**

|  |
| --- |
| function showCategory(category) {  if (category === "Phones") { // 判定 1  displayPhones();  } else if (category === "Laptops") { // 判定 2  displayLaptops();  } else { // 判定 3  displayMonitors();  }  }  content\_copydownload  Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).JavaScript |

**测试用例：**

|  |  |  |
| --- | --- | --- |
| **用例编号** | **分类选项 (category)** | **预期结果** |
| 01 | "Phones" | 显示所有手机商品 |
| 02 | "Laptops" | 显示所有笔记本电脑商品 |
| 03 | "Monitors" | 显示所有显示器商品 |

1. **selenium自动化测试**

测试工具： Selenium 4

测试环境： 平台(Pycharm)、浏览器（Edge）、操作系统（Windows）

功能点：

打开首页

用户注册

用户登录

浏览商品列表

查看商品详情

商品加入购物车

进入购物车页面

删除购物车商品

点击分类页商品

填写订单信息

订单结算功能

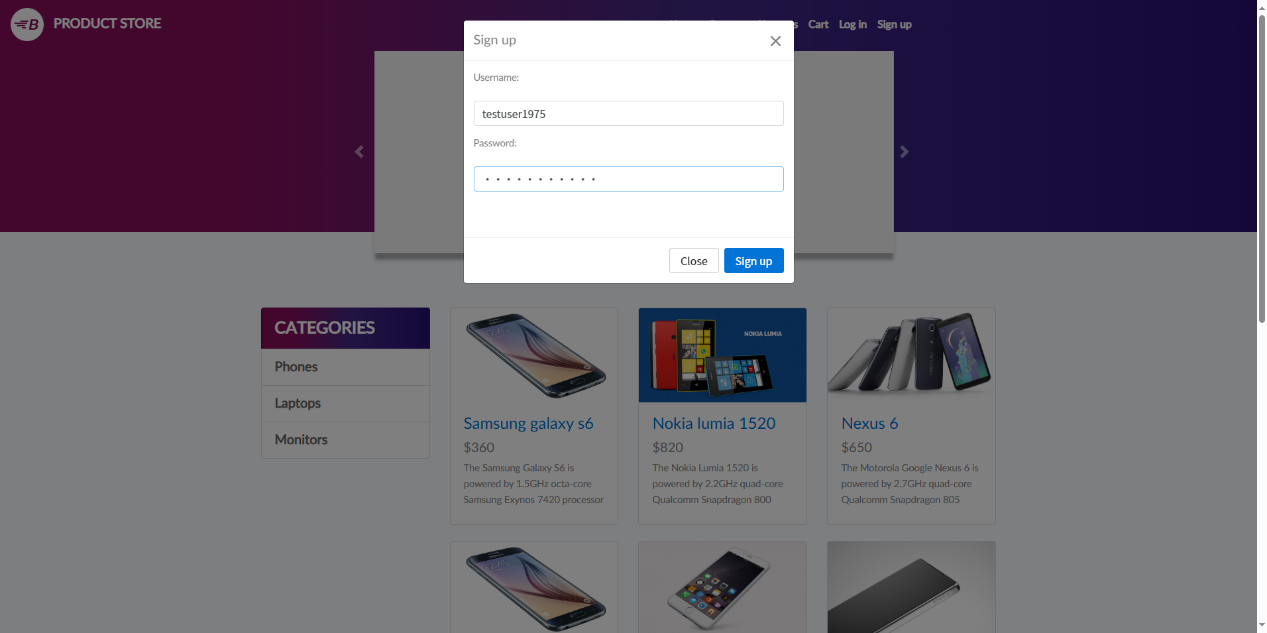
Selenium测试源代码

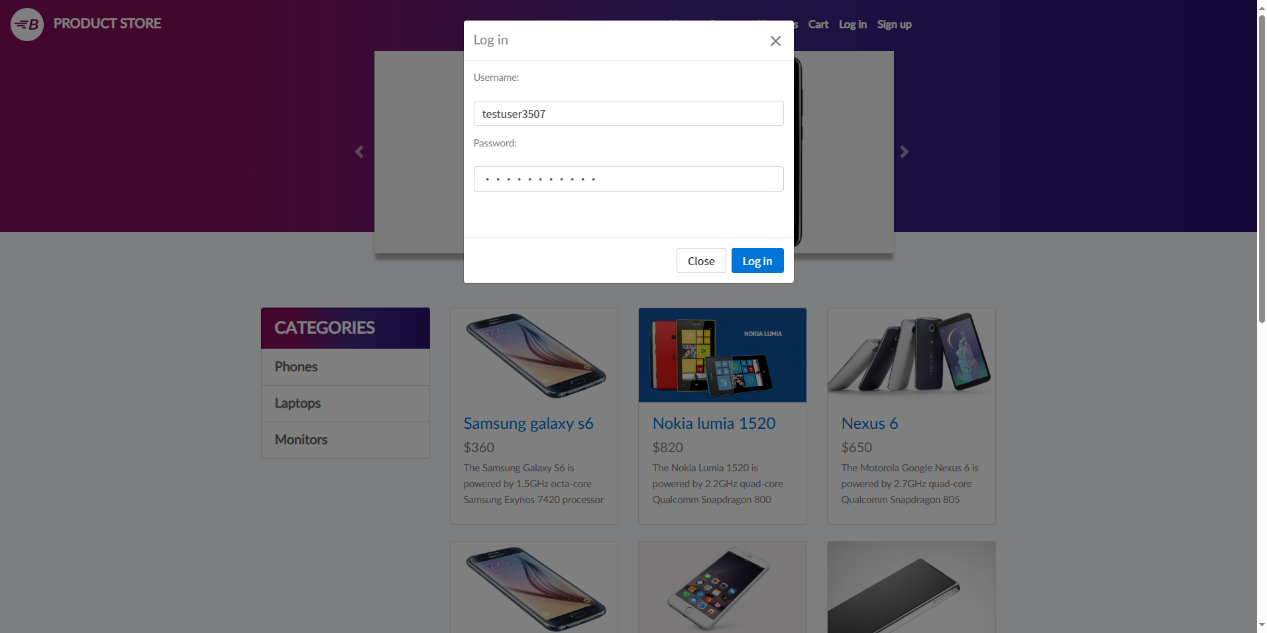
|  |
| --- |
| 测试源代码： |
| from selenium import webdriver  from selenium.webdriver.common.by import By  from selenium.webdriver.support.ui import WebDriverWait  from selenium.webdriver.support import expected\_conditions as EC  from selenium.webdriver.edge.service import Service as EdgeService  import time  import random  # 初始化 Edge 浏览器  service = EdgeService()  driver = webdriver.Edge()  driver.maximize\_window()  # 设置 WebDriverWait 的超时时间  wait = WebDriverWait(driver, 20)  # 生成随机用户名  username = f"testuser{random.randint(1000, 9999)}"  password = "testpass123"  try:  # ==================== 1. 打开首页 ====================  print("1. 打开首页")  driver.get("https://www.demoblaze.com/")  time.sleep(2)  # ==================== 2. 注册功能 ====================  print("2. 开始注册...")  signup\_button = driver.find\_element(By.ID, "signin2")  signup\_button.click()  wait.until(EC.visibility\_of\_element\_located((By.ID, "sign-username")))  username\_field = driver.find\_element(By.ID, "sign-username")  username\_field.send\_keys(username)  password\_field = driver.find\_element(By.ID, "sign-password")  password\_field.send\_keys(password)  register\_button = driver.find\_element(By.XPATH, "//button[text()='Sign up']")  register\_button.click()  time.sleep(2)  try:  alert = driver.switch\_to.alert  print("注册弹窗：", alert.text)  print("注册成功！")  alert.accept()  except:  print("无注册弹窗")  print("注册失败！")  time.sleep(2)  # ==================== 3. 登录功能 ====================  print("3. 开始登录...")  login\_button = driver.find\_element(By.ID, "login2")  login\_button.click()  wait.until(EC.visibility\_of\_element\_located((By.ID, "loginusername")))  loginusername\_field = driver.find\_element(By.ID, "loginusername")  loginusername\_field.send\_keys(username)  loginpassword\_field = driver.find\_element(By.ID, "loginpassword")  loginpassword\_field.send\_keys(password)  login\_submit\_button = driver.find\_element(By.XPATH, "//button[text()='Log in']")  login\_submit\_button.click()  print("登录成功！")  time.sleep(3)  # ==================== 4. 浏览商品功能 ====================  print("4. 浏览商品...")  wait.until(EC.visibility\_of\_element\_located((By.CLASS\_NAME, "card-title")))  products = driver.find\_elements(By.CLASS\_NAME, "card-title")  if products:  print(f"找到{len(products)}个商品，点击第一个商品：{products[0].text}")  products[0].click()  else:  print("未找到商品")  print("商品浏览成功！")  time.sleep(2)  # ==================== 5. 商品加入购物车功能 ====================  print("5. 加入购物车...")  add\_btn = wait.until(EC.element\_to\_be\_clickable((By.XPATH, "//a[text()='Add to cart']")))  add\_btn.click()  time.sleep(2)  try:  alert = driver.switch\_to.alert  print("加入购物车弹窗：", alert.text)  print("加入购物车成功！")  alert.accept()  except:  print("无加入购物车弹窗")  time.sleep(2)  # ==================== 6. 进入购物车页面 ====================  print("6. 进入购物车...")  cart\_icon = driver.find\_element(By.ID, "cartur")  cart\_icon.click()  wait.until(EC.visibility\_of\_element\_located((By.CLASS\_NAME, "success")))  time.sleep(2)  # ==================== 7. 删除购物车商品 ====================  print("7. 删除购物车商品...")  delete\_btns = driver.find\_elements(By.XPATH, "//a[text()='Delete']")  if delete\_btns:  delete\_btns[0].click()  print("已点击删除按钮")  time.sleep(2)  else:  print("购物车无商品可删")  time.sleep(2)  # ==================== 8. 浏览商品分类并添加到购物车 ====================  print("8. 浏览Laptops商品分类并添加到购物车")  driver.find\_element(By.ID, "nava").click()  time.sleep(2)  try:  # 使用 XPath 定位 Laptops 分类链接  laptops\_category\_link = WebDriverWait(driver, 10).until(  EC.element\_to\_be\_clickable((By.XPATH, "//a[@onclick=\"byCat('notebook')\"]"))  )  laptops\_category\_link.click() # 点击Laptops链接  time.sleep(2)  # 获取商品列表  wait.until(EC.visibility\_of\_element\_located((By.CLASS\_NAME, "card-title")))  products = driver.find\_elements(By.CLASS\_NAME, "card-title")  if products:  # 点击第一个商品  products[0].click()  time.sleep(2)  # 加入购物车  add\_btn = wait.until(EC.element\_to\_be\_clickable((By.XPATH, "//a[text()='Add to cart']")))  add\_btn.click()  time.sleep(2)  try:  alert = driver.switch\_to.alert  alert.accept()  print("成功从Laptops分类添加商品到购物车！")  except:  print("该Laptops分类下没有加入购物车按钮")  time.sleep(2)  else:  print("该分类下没有商品")  except Exception as e:  print(f"浏览Laptops商品分类并添加到购物车失败: {e}")  # ==================== 9. 结算功能 ====================  print("9. 测试订单结算功能...")  # 重新添加一个商品  driver.find\_element(By.ID, "nava").click()  time.sleep(2)  products = driver.find\_elements(By.CLASS\_NAME, "card-title")  if products:  products[0].click()  time.sleep(2)  add\_btn = wait.until(EC.element\_to\_be\_clickable((By.XPATH, "//a[text()='Add to cart']")))  add\_btn.click()  time.sleep(2)  try:  alert = driver.switch\_to.alert  alert.accept()  except:  pass  time.sleep(2)  cart\_icon = driver.find\_element(By.ID, "cartur")  cart\_icon.click()  wait.until(EC.visibility\_of\_element\_located((By.CLASS\_NAME, "success")))  time.sleep(2)  # 点击Place Order  place\_order\_button = driver.find\_element(By.XPATH, "//button[text()='Place Order']")  place\_order\_button.click()  wait.until(EC.visibility\_of\_element\_located((By.ID, "orderModalLabel")))  name\_field = driver.find\_element(By.ID, "name")  name\_field.send\_keys("测试用户")  country\_field = driver.find\_element(By.ID, "country")  country\_field.send\_keys("中国")  city\_field = driver.find\_element(By.ID, "city")  city\_field.send\_keys("广州")  card\_field = driver.find\_element(By.ID, "card")  card\_field.send\_keys("1234567890123456")  month\_field = driver.find\_element(By.ID, "month")  month\_field.send\_keys("12")  year\_field = driver.find\_element(By.ID, "year")  year\_field.send\_keys("2025")  time.sleep(2)  purchase\_button = driver.find\_element(By.XPATH, "//button[text()='Purchase']")  purchase\_button.click()  time.sleep(2)  print("订单结算完成！")  else:  print("无商品可结算")  except Exception as e:  print(f"发生错误：{str(e)}")  finally:  print("关闭浏览器...")  driver.quit() |

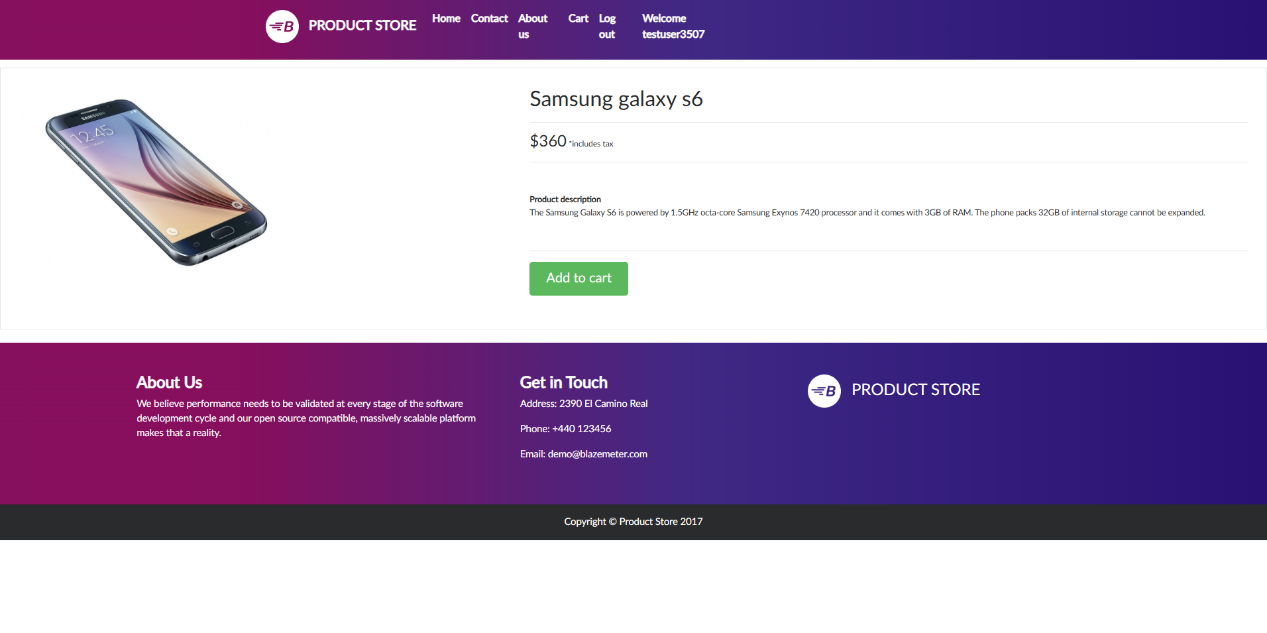
测试结果如下：

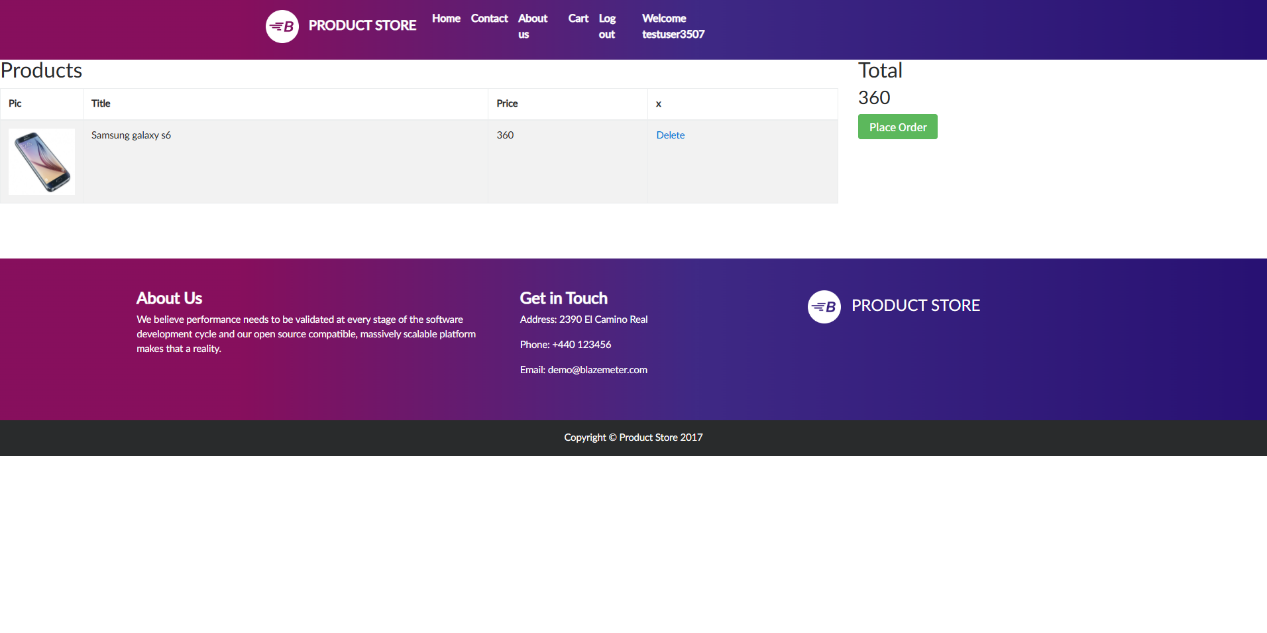
文本

AI 生成的内容可能不正确。









图形用户界面, 应用程序

AI 生成的内容可能不正确。

