Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
им. Н.П. ОГАРЁВА»
(ФГБОУ ВО «МГУ им. Н.П. Огарёва»)

Факультет математики и информационных технологий

Кафедра фундаментальной информатики

ОТЧЁТ ПО ТЕСТИРОВАНИЮ
по дисциплине: Методы тестирования программных продуктов
студента 2 курса магистратуры

Автор отчёта _____В.Е. Родюшкин
подпись, дата

Обозначение работы:

Направление подготовки 02.04.02 Фундаментальная информатика и информационные технологии

Руководитель работы
канд. физ.-мат. наук. _____А.В. Попов
подпись, дата

Оценка

Саранск 2025

Лабораторная работа №3.

Автоматизированное функциональное тестирование
веб-сайта с помощью фреймворка Selenium.

Отчет по тестированию

1. **Описание предмета тестирования**

Wildberries – это российский маркетплейс, где предприниматели могут продавать свои товары, а покупатели данные товары купить. Целью данной работы является написания автотестов для проверки функциональности сайта Wildberries (URL: https://www.wildberries.ru/).

2. **Описание окружения тестирования**

Тип устройства: ноутбук Процессор: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz
Видеокарта: NVIDIA GeForce 930MX
Оперативная память: 8 Gb, DDR3, 2400MHz
Количество ядер: физических 4, логических 8
Операционная система: Windows 10 Домашняя
Разрешение экрана: 1920 × 1080
Антивирус: Kaspersky Free
Расширения: нет
Браузер: Google Chrome 132.0.6834.197 (64 бит)

Характеристики программного обеспечения:

Operating System: Windows 10 (64-bit)

Browser: Google Chrome

Automation Tool: Selenium WebDriver (Python bindings)

Programming Language: Python (Version 3.10)

Libraries: Selenium

IDE: VS Code (for writing and executing scripts)

3. **Use cases (пользовательские сценарии)**

Ниже приведены тест кейсы для Wildberries, как положительные, так и отрицательные сценарии.

Тест кейс 1 (Позитивный): Поиск товара на сайте.

Кейс: Пользователь заходит на сайт и ищет товар через поисковую строку

Шаги:

1. Переход на главную страницу сайта
2. Ввод поискового запроса в поисковую строку сайта (в нашем случае «Кроссовки»)
3. Отправка запроса
4. Получения результатов поиска

Ожидаемый результат: Отображение страницы с результатами поиска по запросу «Кроссовки». На странице есть упоминания «Кроссовки» из запроса

Код тест кейса:

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC import os
from datetime import datetime
import time

# Функция для создания папки, если она не существует
def create_screenshot_folder(folder_name="screenshots"): if not
    os.path.exists(folder_name):
        os.makedirs(folder_name)
        print(f"Папка '{folder_name}' создана.") return
    folder_name

# Функция для сохранения скриншота с временной меткой
```

```
def save_screenshot(browser, step_name, folder_name="screenshots"): timestamp =
    datetime.now().strftime("%Y%m%d_%H%M%S") screenshot_path =
    os.path.join(folder_name,
f"{step_name}_{timestamp}.png")
    browser.save_screenshot(screenshot_path)
    print(f"Скриншот сохранен: {screenshot_path}")

# Основной код
def main():
    # Создание папки для скриншотов
    screenshot_folder = create_screenshot_folder()

    # Инициализация браузера browser =
    webdriver.Chrome() print("Браузер
    запущен.")

    try:
        # Переход на сайт Wildberries
        browser.get("https://www.wildberries.ru/") print("Переход на
        сайт Wildberries выполнен.") time.sleep(5)

                save_screenshot(browser, "homepage", screenshot_folder)  #
 Скриншот главной страницы

        # Поиск поля ввода и ввод текста
        search_box = browser.find_element(By.ID, "searchInput")
        search_box.send_keys("Кроссовки")
        print("Текст 'Кроссовки' введен в поле поиска.")
        time.sleep(2)
        save_screenshot(browser, "search_input", screenshot_folder)  #
Скриншот с введенным текстом
        search_box.send_keys(Keys.RETURN)
        print("Запрос отправлен.")

        # Ожидание появления результатов поиска print("Ожидание
        появления результатов поиска...") WebDriverWait(browser,
        15).until(
            EC.presence_of_element_located((By.XPATH,
"//*[contains(text(), 'Кроссовки')]"))
        )
        print("Результаты поиска успешно загружены.")
        time.sleep(2)
        save_screenshot(browser, "search_results", screenshot_folder)  #
Скриншот результатов поиска

        # Проверка наличия текста "Кроссовки" в исходном коде страницы
        assert "Кроссовки" in browser.page_source
```

```
            print("Текст 'Кроссовки' найден на странице.")

        except Exception as e:
            print("Ошибка:", e)
            save_screenshot(browser, "error", screenshot_folder)  # Скриншот в случае ошибки

        finally:
            # Закрытие браузера
            browser.quit() print("Браузер
            закрыт.")

# Запуск основного кода
if __name__ == "__main__":
    main()
```

Сохраненные при выполнении теста скриншоты:

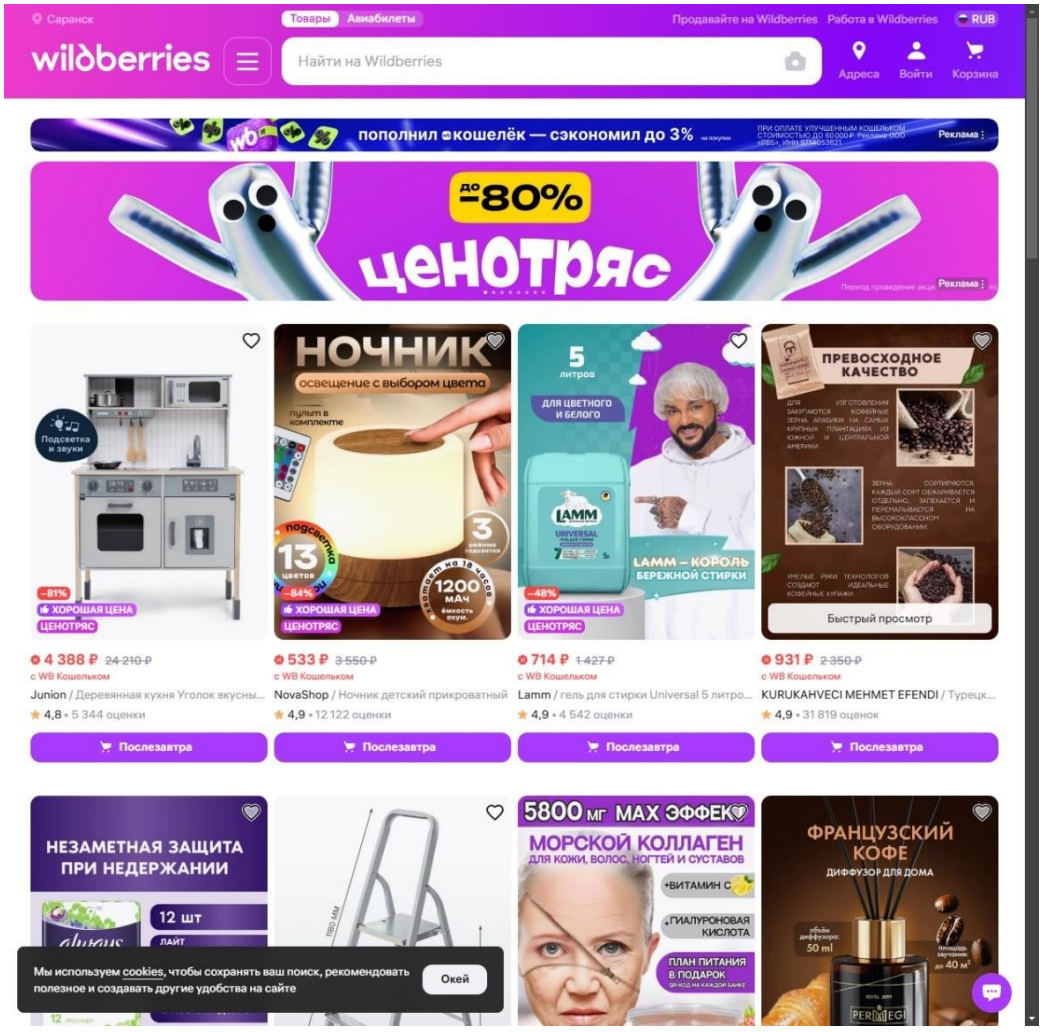

Рисунок 1 – Главная страница сайта

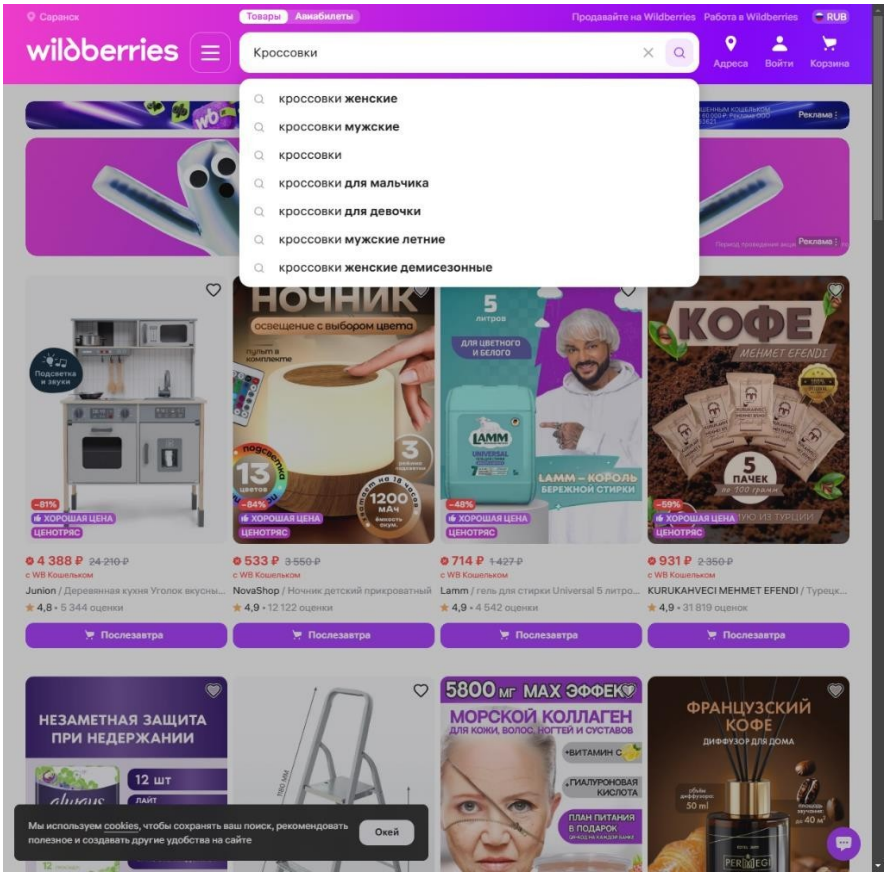

Рисунок 2 – Ввод в поисковую строку запроса «Кроссовки»

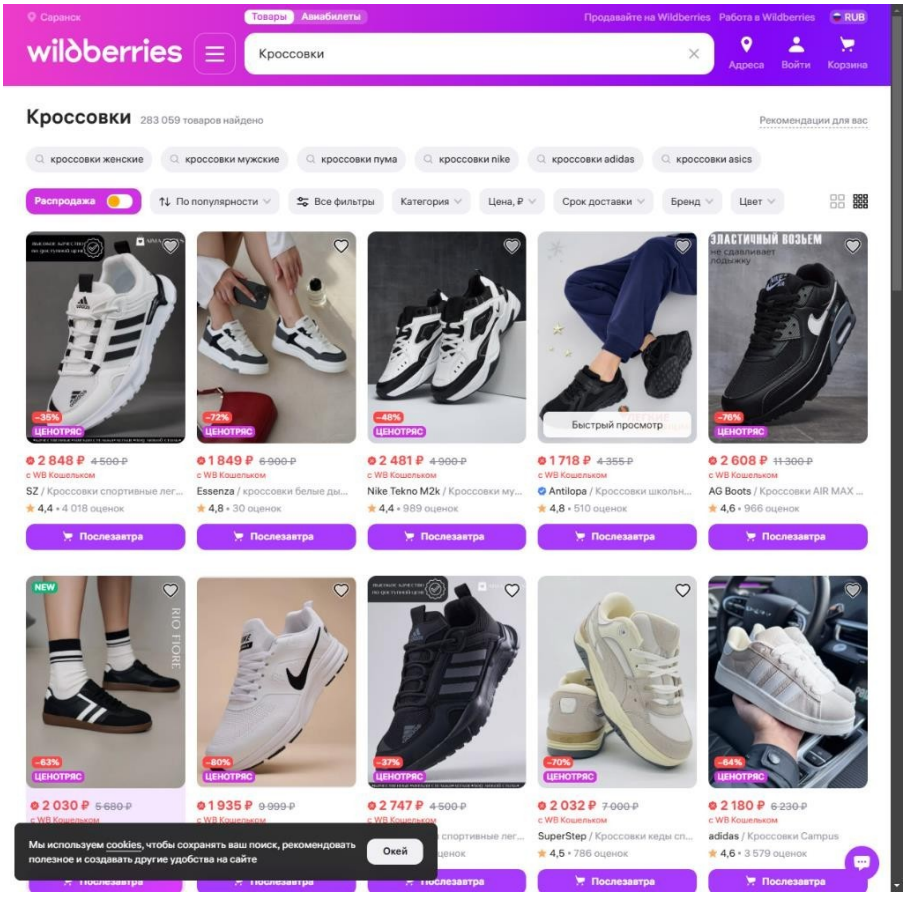

Рисунок 3 – Результат поиска

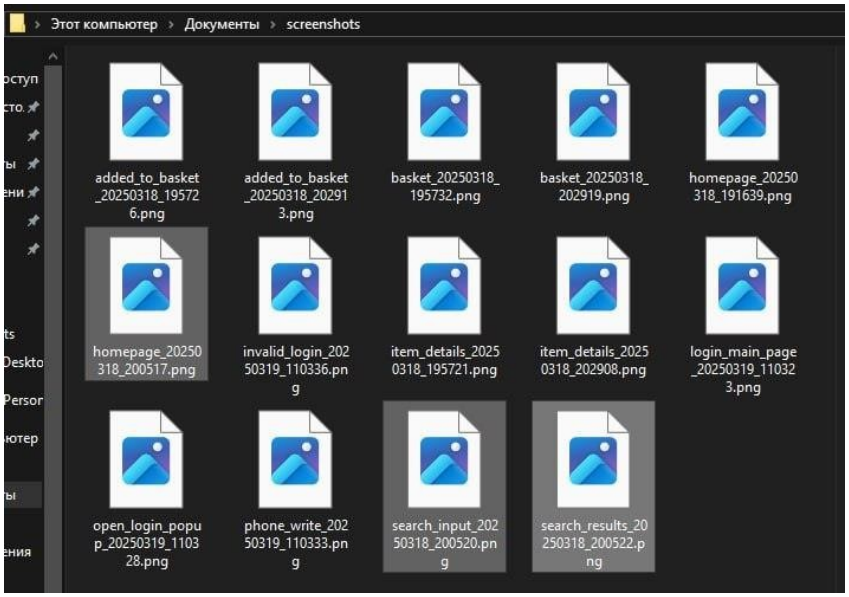

Рисунок 4 – Сохраненные скриншоты теста в папке screenshots

```
PS C:\Users\nikit\Documents> python .\wb_selenium.py
Папка 'screenshots' создана.

DevTools listening on ws://127.0.0.1:56081/devtools/browser/58eae882-e0ec-49a3-b949-bf6bbbfd0e5f
Браузер запущен.
Переход на сайт Wildberries выполнен.
Скриншот сохранен: screenshots\homepage_20250318_191639.png
Текст 'Кроссовки' введен в поле поиска.
Скриншот сохранен: screenshots\search_input_20250318_191639.png
Запрос отправлен.
Ожидание появления результатов поиска...
Результаты поиска успешно загружены.
Скриншот сохранен: screenshots\search_results_20250318_191642.png
Текст 'Кроссовки' найден на странице.
Браузер закрыт.
```

Рисунок 5 – Консоль после выполнения теста

Тест кейс 2 (Позитивный): Добавление товара в корзину.

Кейс: Пользователь заходит на страницу товара (который есть в наличии) и хочет добавить его в корзину.

Шаги:

1. Переход на страницу товара
2. Добавление товара в корзину.
3. Переход в корзину пользователя

Ожидаемый результат: Товар успешно добавлен в корзину. В корзине пользователя есть данный товар.

Код тест кейса:

```
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 import
os
from datetime import datetime
import time

# Функция для создания папки, если она не существует
defcreate_screenshot_folder(folder_name="screenshots"): if not
    os.path.exists(folder_name):
        os.makedirs(folder_name)
        print(f"Папка '{folder_name}' создана.")
    return folder_name

# Функция для сохранения скриншота с временной меткой
def save_screenshot(browser, step_name, folder_name="screenshots"): timestamp =
    datetime.now().strftime("%Y%m%d_%H%M%S") screenshot_path =
    os.path.join(folder_name,
f"{step_name}_{timestamp}.png")
    browser.save_screenshot(screenshot_path)
    print(f"Скриншот сохранен: {screenshot_path}")

# Основной код
def main():
    # Создание папки для скриншотов
    screenshot_folder = create_screenshot_folder()

    # Инициализация браузера browser =
    webdriver.Chrome() print("Браузер
    запущен.")

    try:
        # Переход на сайт Wildberries
        browser.get("https://www.wildberries.ru/catalog/278377729/detail.
aspx")
        print("Переход на сайт Wildberries выполнен.")
        time.sleep(5)
        save_screenshot(browser, "item_details", screenshot_folder)  #
Скриншот страницы товара

        # Ожидание и клик по кнопке "Добавить в корзину"
```

```
            try:
                add_to_cart_button = WebDriverWait(browser, 15).until(
                    EC.element_to_be_clickable((By.XPATH,
"//button[contains(@class, 'order_button') and contains(., 'Добавить в корзину')]"))
                )
                add_to_cart_button.click() print("Товар
                добавлен в корзину.") time.sleep(5)
                save_screenshot(browser, "added_to_basket", screenshot_folder)  # Скриншот
добавленного в корзину товара
            except Exception as e:
                print("Ошибка при добавлении товара в корзину:", e)
                save_screenshot(browser,"error_add_to_cart", screenshot_folder)

            # Переход в корзину
            try:
                cart_button = WebDriverWait(browser, 15).until(
                    EC.element_to_be_clickable((By.CLASS_NAME,"navbar-
  pc_item.j-item-basket"))
                )
                cart_button.click()
                print("Переход в корзину выполнен.")
                time.sleep(5)
                # Ожидание появления товара в корзине
                WebDriverWait(browser, 15).until(
                    EC.presence_of_element_located((By.XPATH,
"//*[contains(text(), 'Видеокарта')]"))
                )
                save_screenshot(browser, "basket", screenshot_folder)  #
Скриншот корзины

                # Проверка наличия текста "Видеокарта" на странице assert
                "Видеокарта" in browser.page_source print("Текст 'Видеокарта'
                найден на странице.")
            except Exception as e:
                print("Ошибка при переходе в корзину или проверке товара:",
e)
                save_screenshot(browser, "error_basket", screenshot_folder)

        except Exception as e:
            print("Общая ошибка:", e)
            save_screenshot(browser, "error", screenshot_folder)  # Скриншот в случае ошибки

        finally:
            # Закрытие браузера
```

```
            browser.quit()
            print("Браузер закрыт.")

# Запуск основного кода
if __name__ == "__main__":
    main()
```

Сохраненные при выполнении теста скриншоты:

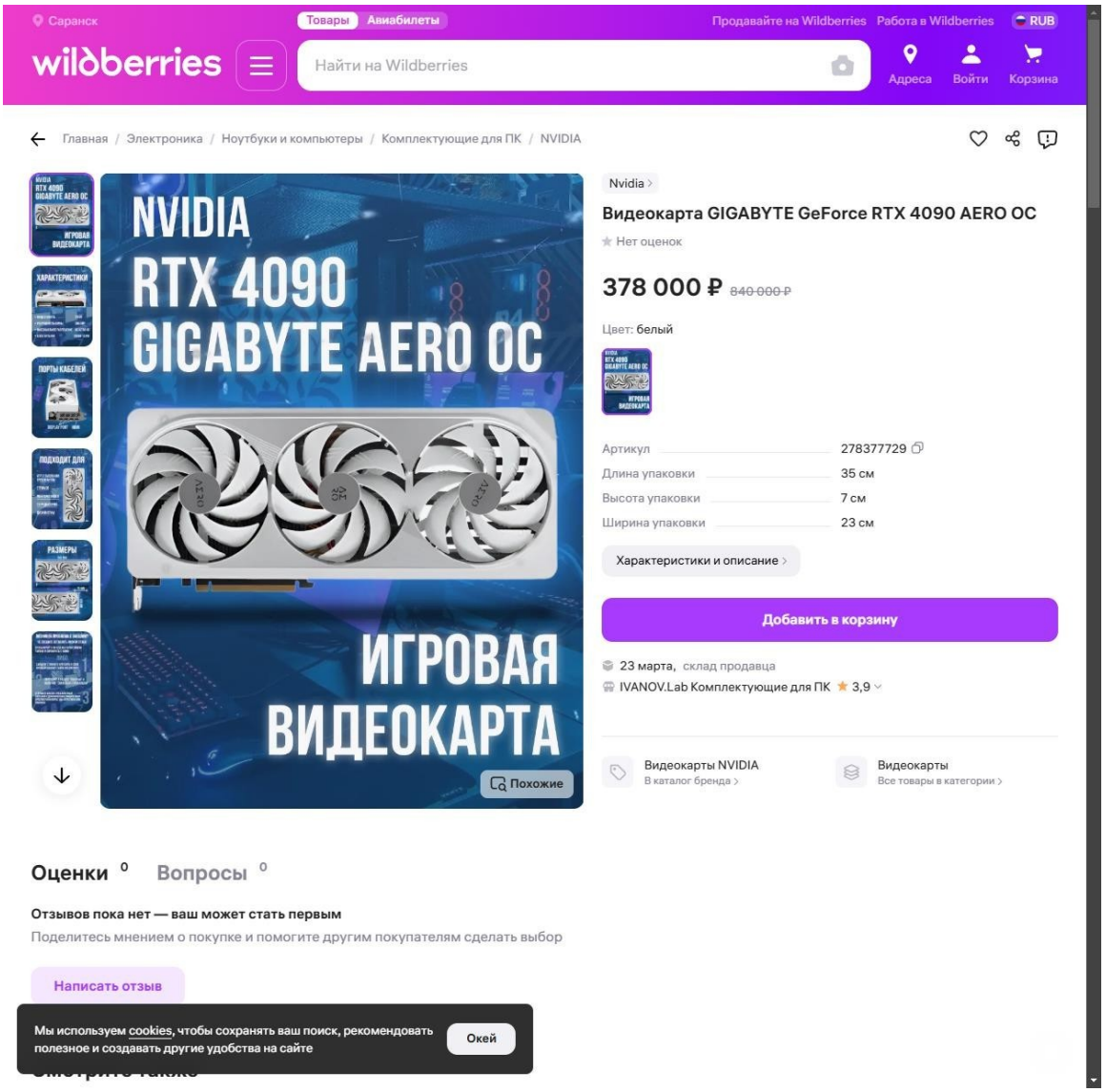

Рисунок 6 – Страница товара

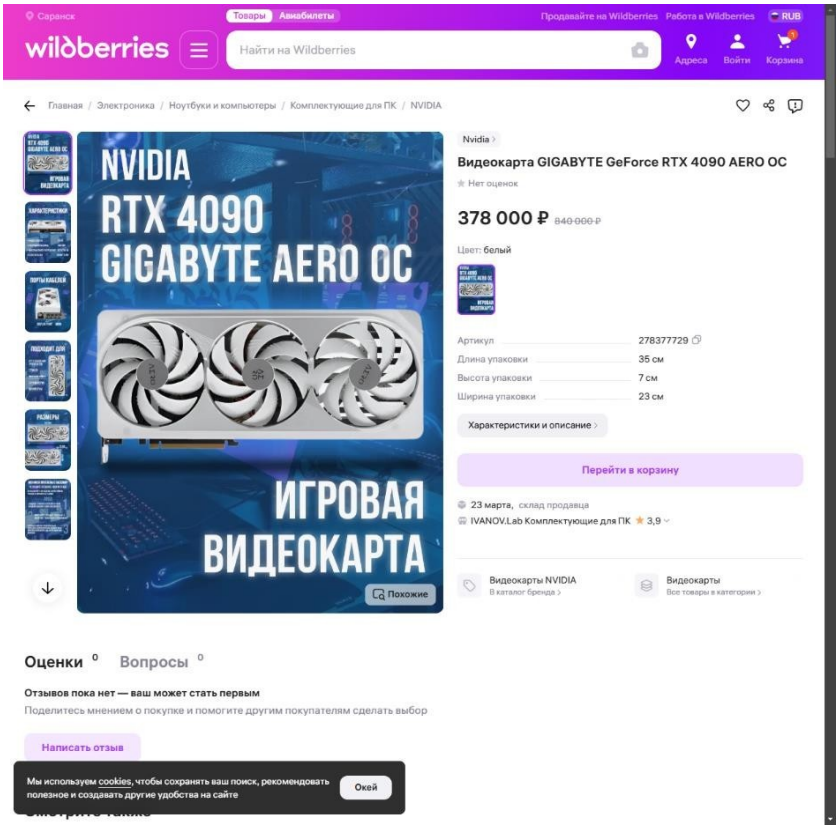

Рисунок 7 – Товар добавлен в корзину

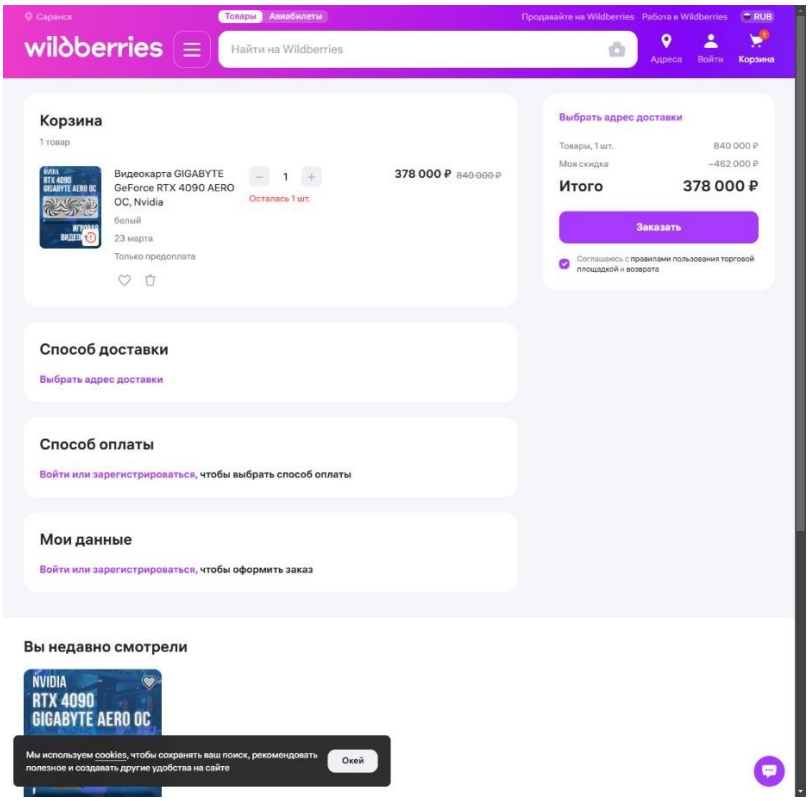

Рисунок 8 – Корзина

```
PS C:\Users\nikit\Documents> python .\wb_selenium.py

DevTools listening on ws://127.0.0.1:58345/devtools/browser/f3127c67-3dc6-45f8-b13c-9bc6817b1785
Браузер запущен.
Переход на сайт Wildberries выполнен.
Скриншот сохранен: screenshots\item_details_20250318_195721.png
Товар добавлен в корзину.
Скриншот сохранен: screenshots\added_to_basket_20250318_195726.png
Переход в корзину выполнен.
Скриншот сохранен: screenshots\basket_20250318_195732.png
Текст 'Видеокарта' найден на странице.
Браузер закрыт.
```

Рисунок 9 – Консоль после выполнения теста

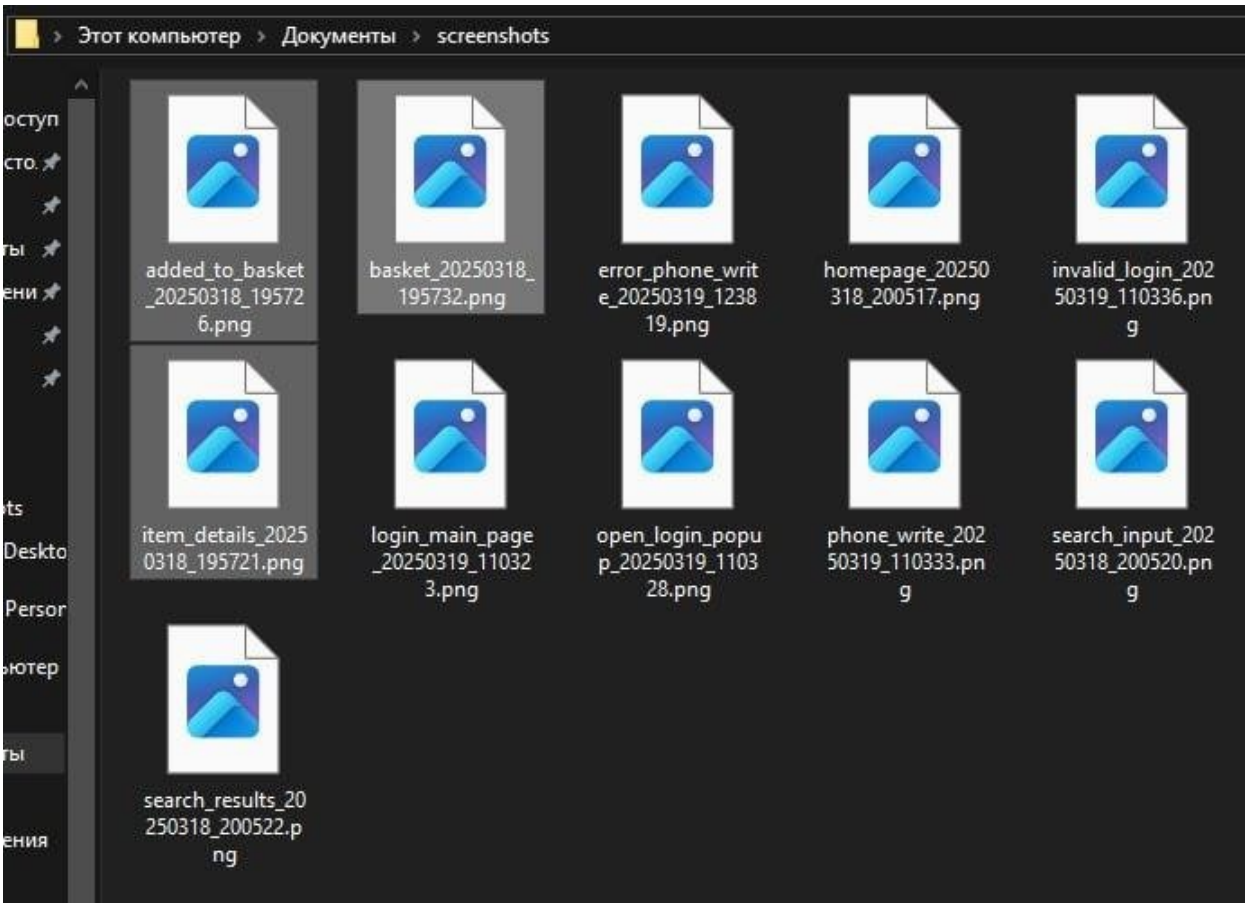

Рисунок 10 – Сохраненные скриншоты теста в папке screenshots

Тест кейс 3 (Негативный): Вход в профиль.

Кейс: Пользователь заходит на сайт, хочет войти в профиль и вводит некорректный номер телефона.

Шаги:

1. Переход на сайт
2. Нажатие кнопки «Войти».
3. Ввод некорректного номера телефона в модальном окне
4. Нажатие кнопки «Получить код»

Ожидаемый результат: Отображается сообщение об ошибке.

Код тест кейса:

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC import os
from datetime import datetime
import time

# Функция для создания папки, если она не существует
defcreate_screenshot_folder(folder_name="screenshots"): if not
    os.path.exists(folder_name):
        os.makedirs(folder_name)
        print(f"Папка '{folder_name}' создана.") return
    folder_name

# Функция для сохранения скриншота с временной меткой
def save_screenshot(browser, step_name, folder_name="screenshots"): timestamp =
    datetime.now().strftime("%Y%m%d_%H%M%S") screenshot_path =
    os.path.join(folder_name,
f"{step_name}_{timestamp}.png")
    browser.save_screenshot(screenshot_path)
    print(f"Скриншот сохранен: {screenshot_path}")

# Основной код
def main():
    # Создание папки для скриншотов
    screenshot_folder = create_screenshot_folder()

    # Инициализация браузера browser =
    webdriver.Chrome() print("Браузер
    запущен.")

    try:
        # Переход на сайт Wildberries
        browser.get("https://www.wildberries.ru") print("Переход на сайт
        Wildberries выполнен.") time.sleep(5)
```

```
            save_screenshot(browser, "login_main_page", screenshot_folder)  # Скриншот страницы товара

            # Ожидание и клик по кнопке "Войти"
            try:
                login_button = WebDriverWait(browser, 15).until(
                    EC.element_to_be_clickable((By.CLASS_NAME, "navbar-pc_link.j-main-login"))
                )
                login_button.click()
                print("Переход в модальное окно логина выполнен.")
                time.sleep(5)
                save_screenshot(browser,"open_login_popup", screenshot_folder)  # Скриншот открытой модалки
            except Exception as e:
                print("Ошибка при попытке открыть окно логина:", e)
                save_screenshot(browser,"error_add_to_cart", screenshot_folder)

            # Ищем инпут для ввода телефона и вводим некорректный номер
            try:
                tel_input = browser.find_element(By.CLASS_NAME, "input-item") # Передаем некорректный номер tel_input.send_keys("999222334")
                print("Значение номера телефона прописано.")
                time.sleep(5)
                save_screenshot(browser, "phone_write", screenshot_folder)  # Скриншот введенного номера
                print("Ошибка при попытке открыть окно логина:", e) except Exception as e:
                print("Ошибка при попытке прописать номер телефона:", e)
                save_screenshot(browser,"error_phone_write", screenshot_folder)

            # Пытаемся войти по невалидному номеру
            request_button = browser.find_element(By.ID, "requestCode")
            request_button.click()
            print("Попытка войти произошла")
            time.sleep(2)
            error_message = browser.find_element(By.CLASS_NAME, "j-error- full-phone.field-validation-error")
            assert error_message.is_displayed()
            assert "Некорректный формат номера" in browser.page_source
            save_screenshot(browser, "invalid_login", screenshot_folder)
            print("Ошибка неправильнного ввода номера телефона отобразилась")
        except Exception as e:
            print("Ошибка:", e)
```

```
            save_screenshot(browser, "error", screenshot_folder)  # Скриншот в случае ошибки

    finally:
        # Закрытие браузера
        browser.quit() print("Браузер
        закрыт.")

# Запуск основного кода
if __name__ == "__main__":
    main()
```

Сохраненные при выполнении теста скриншоты:

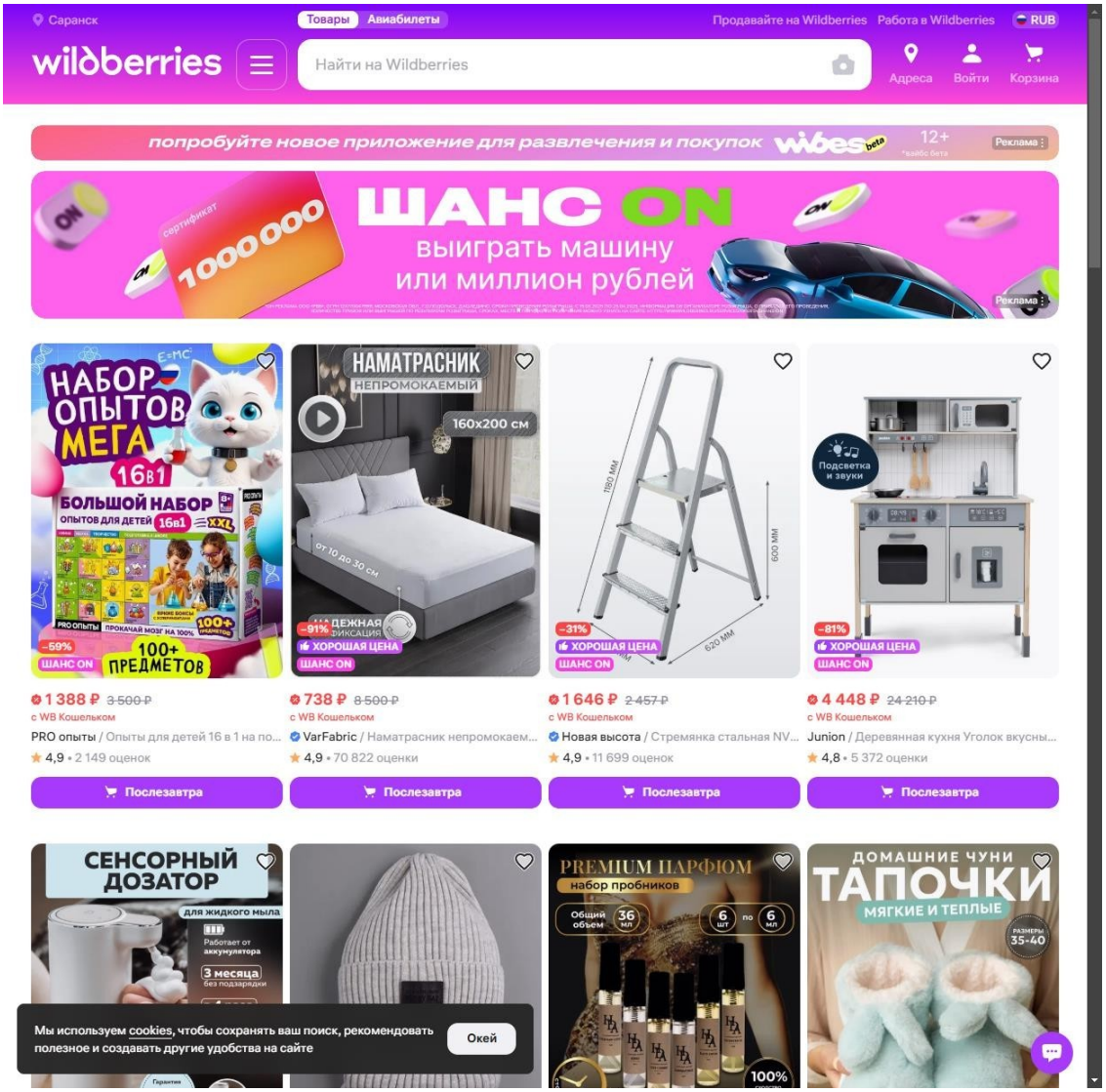

Рисунок 11 – Главная страница

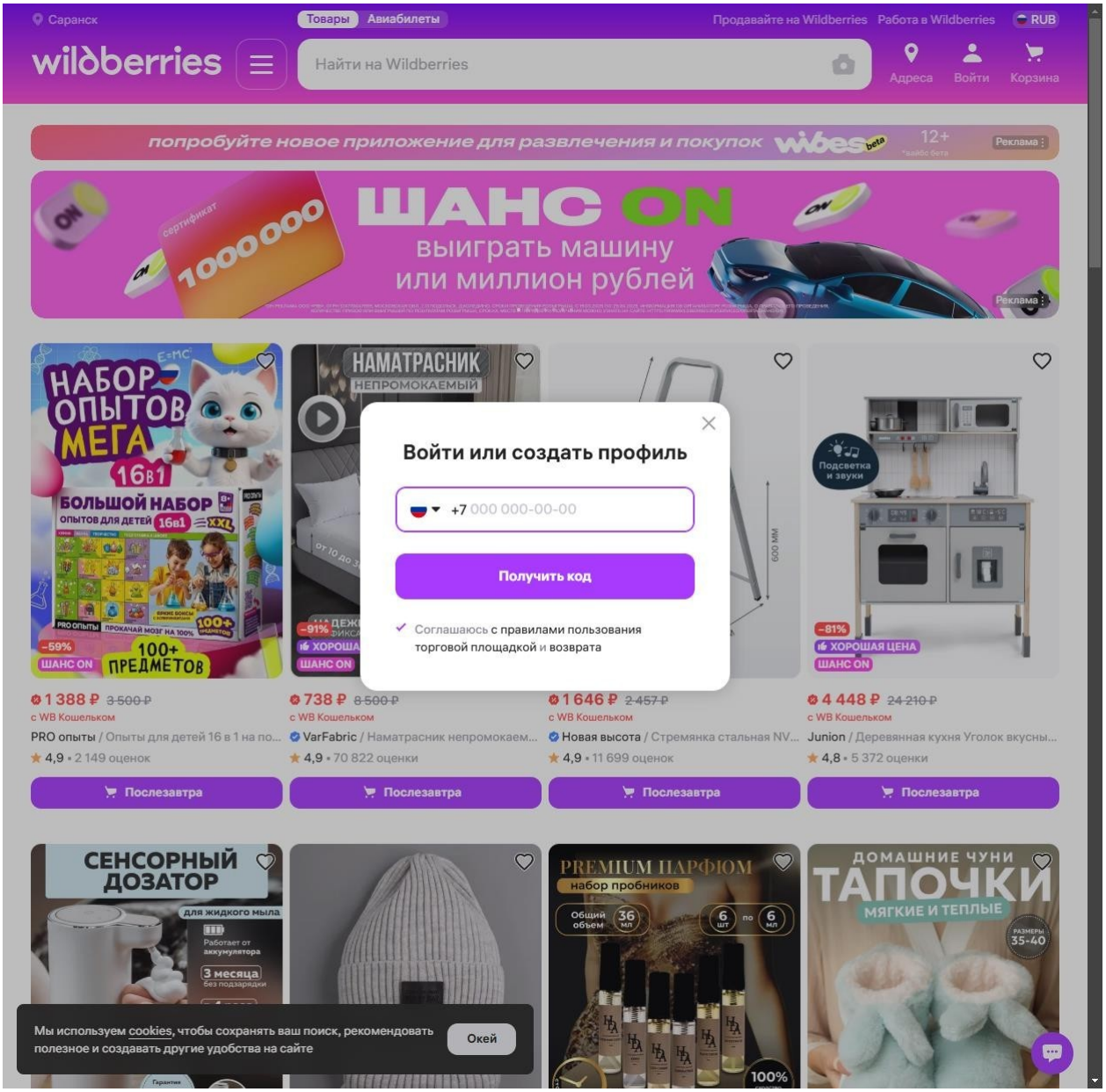

Рисунок 12 – Окно, после нажатия кнопки «Войти»

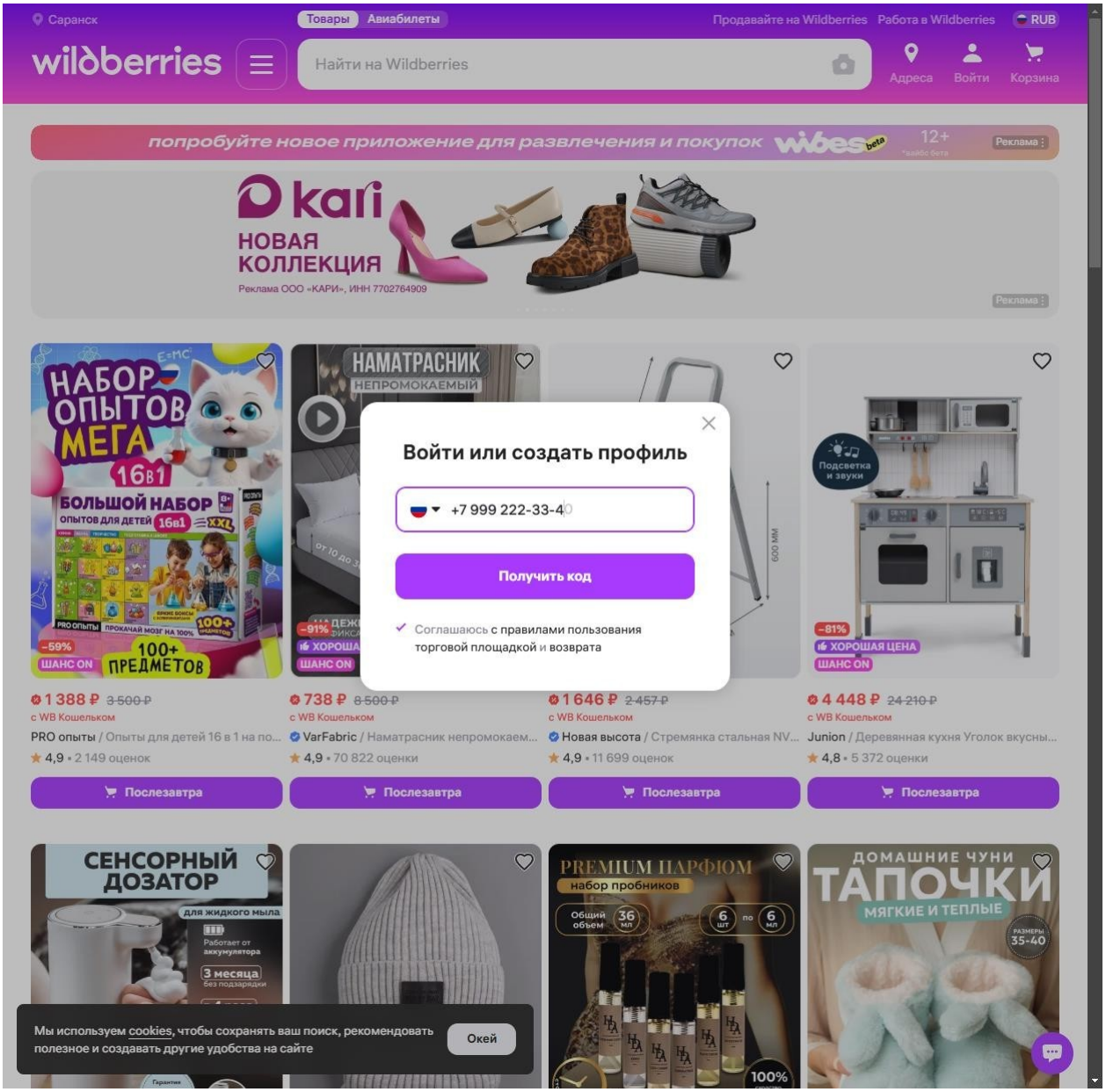

Рисунок 13 – Ввод некорректного номера

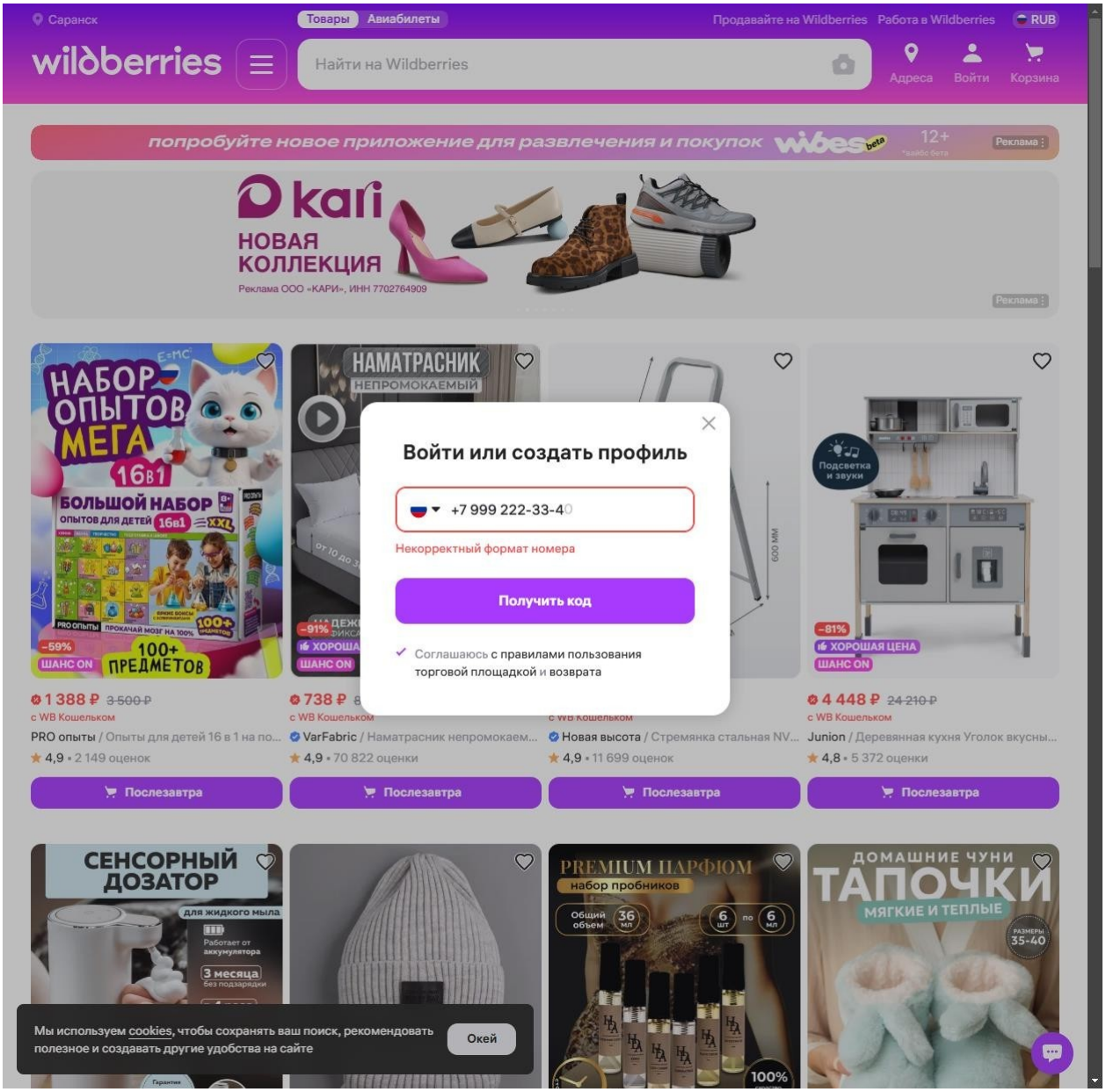

Рисунок 14 – Результат, после нажатия кнопки «Получить код»

```
PS C:\Users\nikit\Documents> python .\wb_selenium.py

DevTools listening on ws://127.0.0.1:50102/devtools/browser/92c6fd3f-c831-4695-a7c8-2a99198485f6
Браузер запущен.
Переход на сайт Wildberries выполнен.
Скриншот сохранен: screenshots\login_main_page_20250319_110323.png
Переход в модальное окно логина выполнен.
Скриншот сохранен: screenshots\open_login_popup_20250319_110328.png
Значение номера телефона прописано.
Created TensorFlow Lite XNNPACK delegate for CPU.
Скриншот сохранен: screenshots\phone_write_20250319_110333.png
Ошибка при попытке прописать номер телефона: cannot access local variable 'e' where it is not associated with a value
Скриншот сохранен: screenshots\error_phone_write_20250319_110333.png
Попытка войти произошла
Скриншот сохранен: screenshots\invalid_login_20250319_110336.png
Ошибка неправильнного ввода номера телефона отобразилась
Браузер закрыт.
```

Рисунок 15 – Консоль результата работы теста

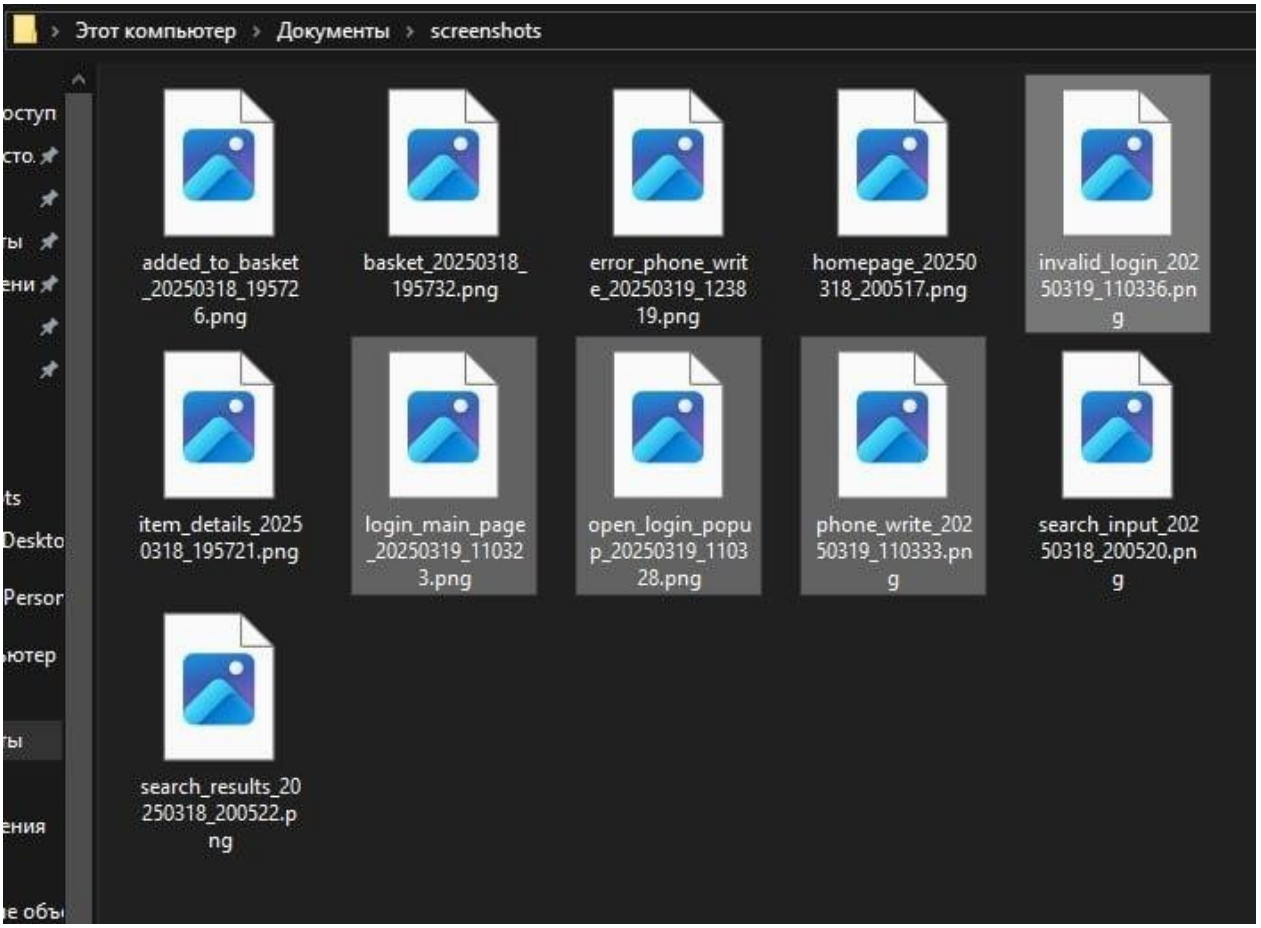

Рисунок 16 – Сохраненные скриншоты теста в папке screenshots