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

Автор отчёта по лабораторной работе _____Д.С. Перепелкин
подпись, дата

Обозначение работы

Направление подготовки 02.04.02 Фундаментальная информатика и информационные технологии

Руководитель работы,
канд. тех. наук _____А.В. Попов
подпись дата

Саранск 2025

Лабораторная работа №3.

Автоматизированное функциональное тестирование
веб-сайта с помощью фреймворка Selenium.

1. **Описание предмета тестирования**

«Кинопо́иск» (Кинопоиск HD) – русскоязычный интернет-сервис с условно свободно редактируемой базой данных и интернет-издание о кинематографе. С 2018 года – онлайн-кинотеатр. С 2020 года совместно с «Плюс Студией», продюсерским центром «Яндекса», производит и распространяет оригинальные фильмы и сериалы (URL: https://hd.kinopoisk.ru).

2. **Описание окружения тестирования**

Тип устройства: ноутбук

Процессор: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz

Видеокарта встроенная: Intel Iris Xe Graphics

Оперативная память: 8,00 ГБ

Тип системы: 64-разрядная операционная система, процессор x64

Количество ядер: физических 4

Операционная система: Windows 11 Домашняя

Разрешение экрана: 2160 x 1440

Антивирус: встроенный защитник Windows

Расширения: нет

Браузер: Google Chrome 133.0.6943.142 (Официальная сборка), (64 бит)

Характеристики программного обеспечения:

Operating System: Windows 11 (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 Кинопоиска, как положительные, так и отрицательные сценарии.

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

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

Шаги:

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

Ожидаемый результат: Отображение страницы с результатом поиска по запросу «Триггер». На странице появится список с фильмами по названию «Триггер».

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

options = webdriver.ChromeOptions()
options.add_argument("user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36")

driver = webdriver.Chrome(options=options)


# 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()  # Можно заменить на Firefox, Edge и т. д.
    print("Браузер запущен.")

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

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

        # Ожидание появления поисковой строки
        search_box = WebDriverWait(browser, 10).until(
            EC.presence_of_element_located((By.NAME, "kp_query"))
        )

        # Ввод поискового запроса
        search_box.send_keys("Триггер")
        search_box.send_keys(Keys.RETURN)
        print("Поисковый запрос отправлен.")

        # Ожидание загрузки результатов поиска
        WebDriverWait(browser, 10).until(
            EC.presence_of_element_located((By.CLASS_NAME,
"styles_root__ti07r"))  # Возможно, нужно обновить селектор
        )

        save_screenshot(browser, "search_results", screenshot_folder)  # Скриншот
страницы с результатами поиска
        print("Результаты поиска отображены.")

    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:

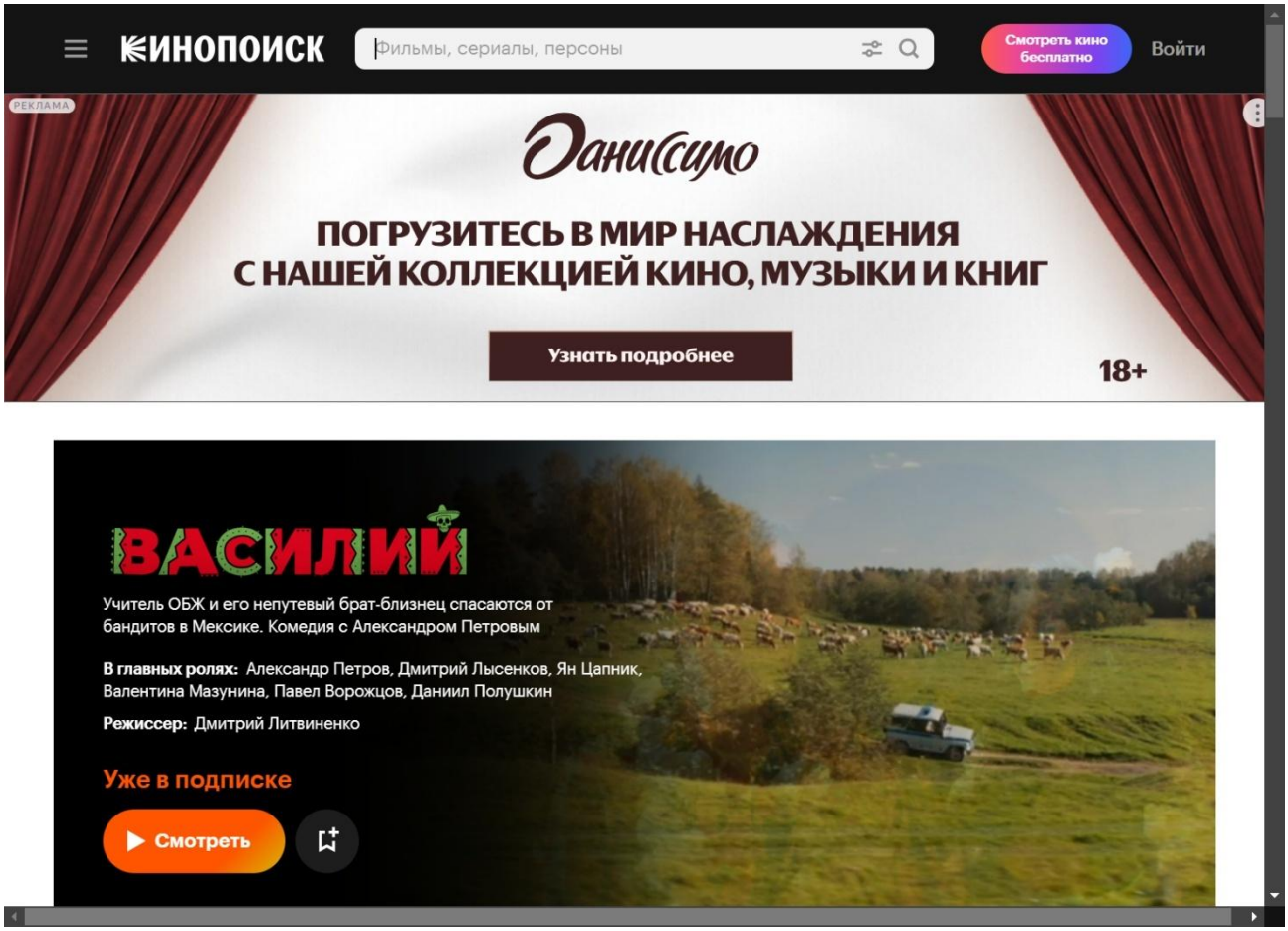

Рисунок 1 – Главная страница сайта

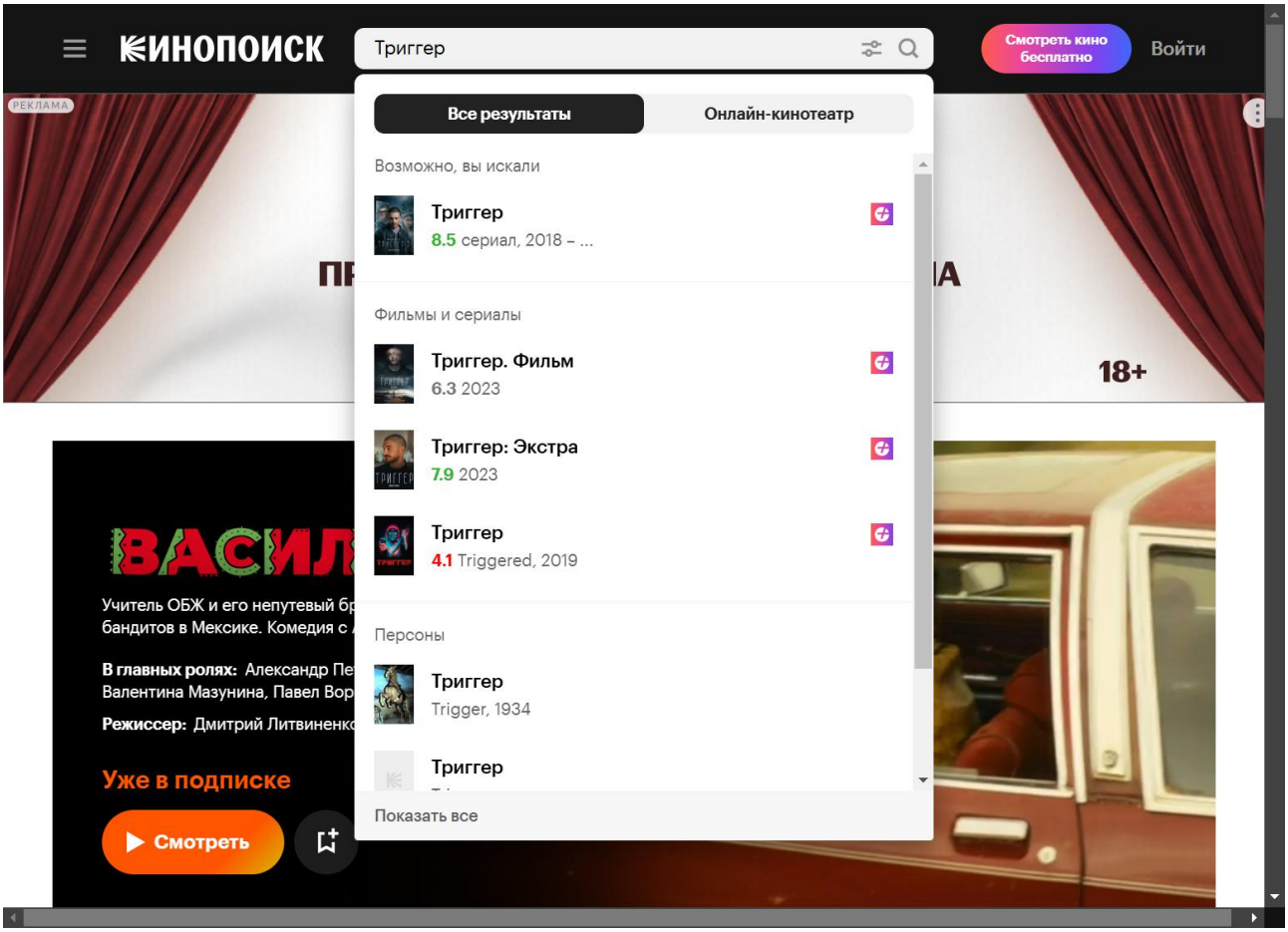

Рисунок 2 – Ввод в поисковую строку запроса «Триггер»

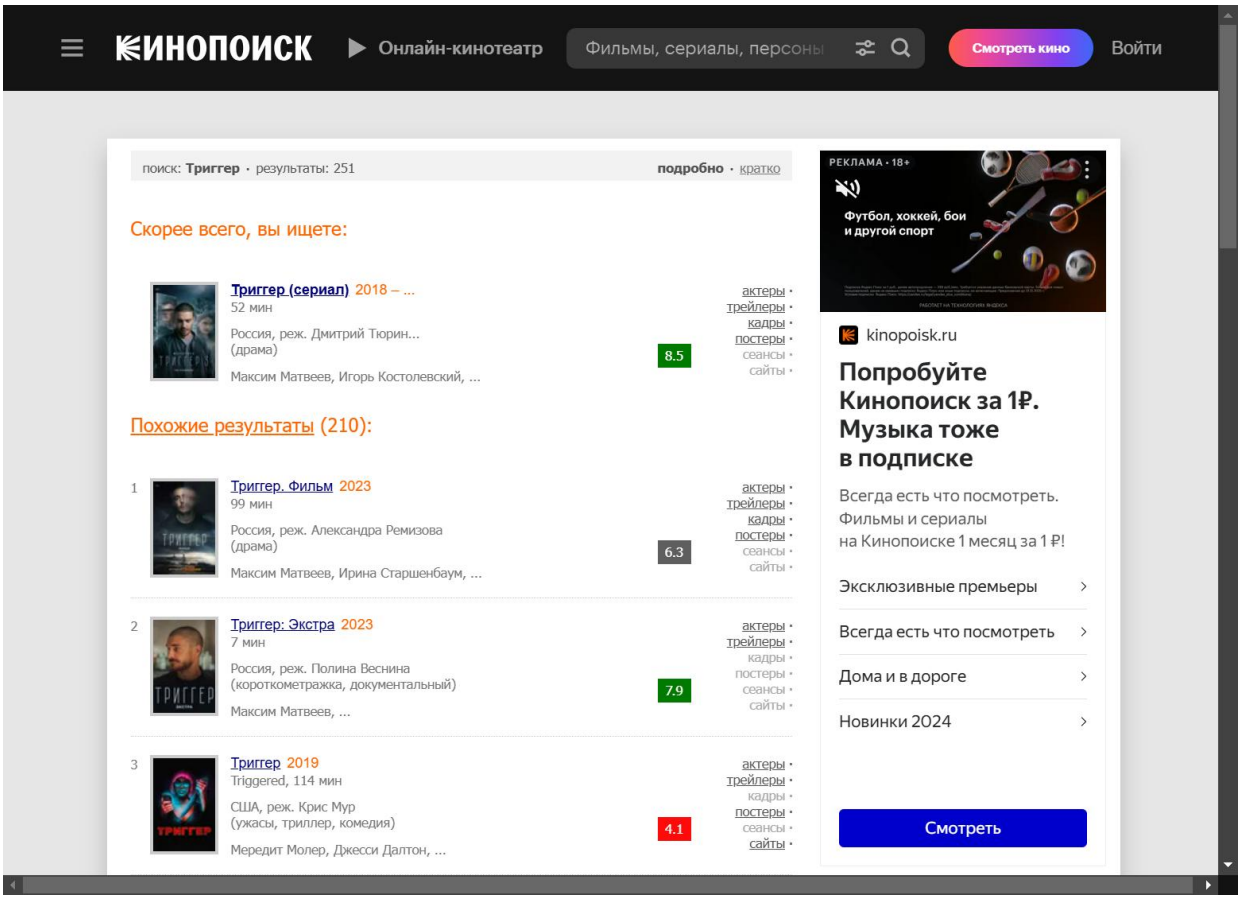

Рисунок 3 – Результат поиска

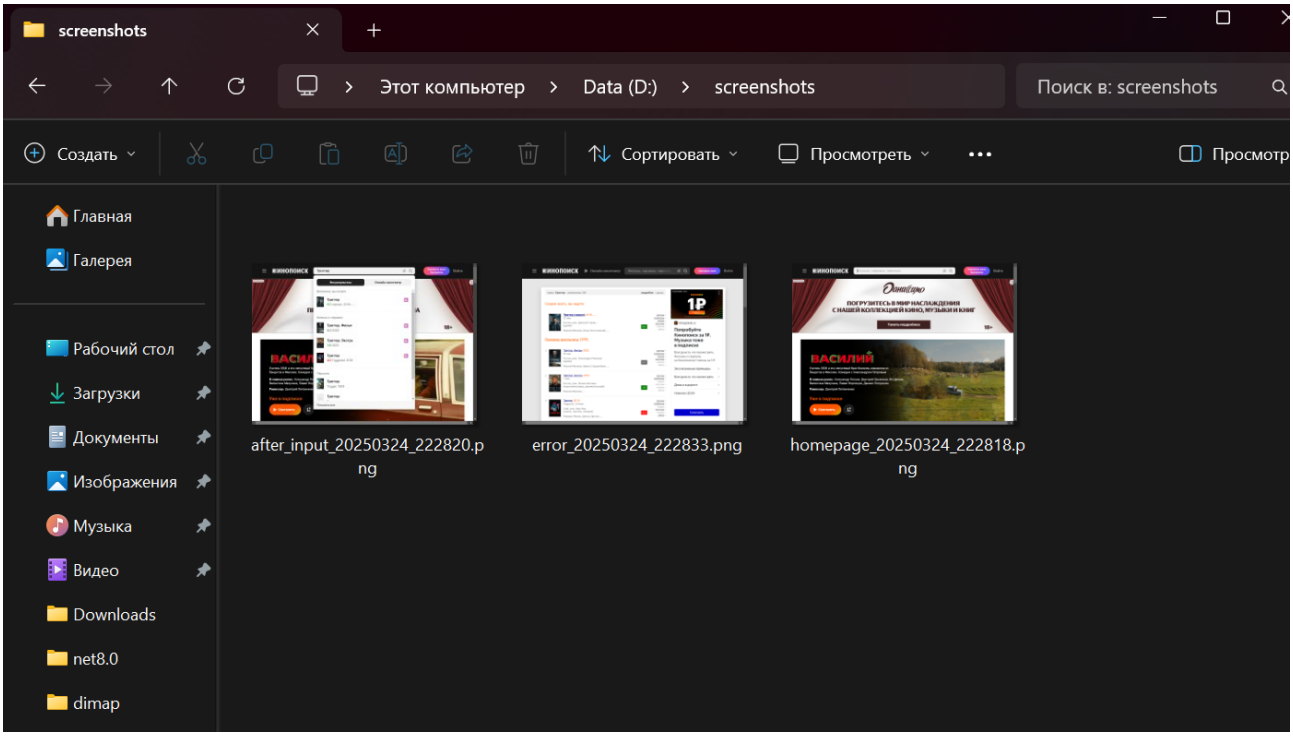

Рисунок 4 – Сохраненные скриншоты теста в папке screenshots

Тест кейс 2 (Позитивный): Открытие страницы фильма.

Кейс: Пользователь ищет фильм и открывает его страницу.

Шаги:

1. Открыть Кинопоиск
2. Ввести в поиск название фильма ("Триггер").
3. Отправить запрос.
4. Кликнуть на первый результат из списка.
5. Дождаться загрузки страницы фильма.

Ожидаемый результат: Страница фильма открыта, отображается заголовок и информация о фильме.

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 для скриншотов
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
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}")

# Настройки браузера
options = webdriver.ChromeOptions()
options.add_argument("user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36")

# Запуск браузера
driver = webdriver.Chrome(options=options)

try:
    screenshot_folder = create_screenshot_folder()

    driver.get("https://www.kinopoisk.ru/")
    time.sleep(5)
    input("Пройди капчу вручную и нажми Enter...")
    save_screenshot(driver, "homepage", screenshot_folder)



    search_box = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.NAME, "kp_query"))
    )
    search_box.send_keys("Триггер")
    time.sleep(2)
    save_screenshot(driver, "after_input", screenshot_folder)

    search_box.send_keys(Keys.RETURN)

    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.CLASS_NAME, "styles_root__ti07r"))  #
Проверь актуальный класс
    )
    save_screenshot(driver, "search_results", screenshot_folder)

    first_result = WebDriverWait(driver, 10).until(
        EC.presence_of_all_elements_located((By.XPATH, "//a[contains(@href,
'/film/')]"))
    )
    print(f" Количество найденных элементов: {len(first_result)}")

    if len(first_result) > 0:
        first_result[0].click()  # Кликаем по первому результату
```

```
        print(" Кликаем по первому результату.")
    else:
        save_screenshot(driver, "no_results", screenshot_folder)

    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.CLASS_NAME, "styles_title__3s2l3"))  # 
Проверь актуальный селектор!
    )
    time.sleep(2)
    save_screenshot(driver, "movie_page", screenshot_folder)
    print(" Страница фильма открыта.")

except Exception as e:
    print(" Общая ошибка:", e)
    save_screenshot(driver, "error", screenshot_folder)

finally:
    if driver:
        print(" Браузер закрыт.")
        driver.quit()
```

Сохраненные при выполнении теста скриншоты:

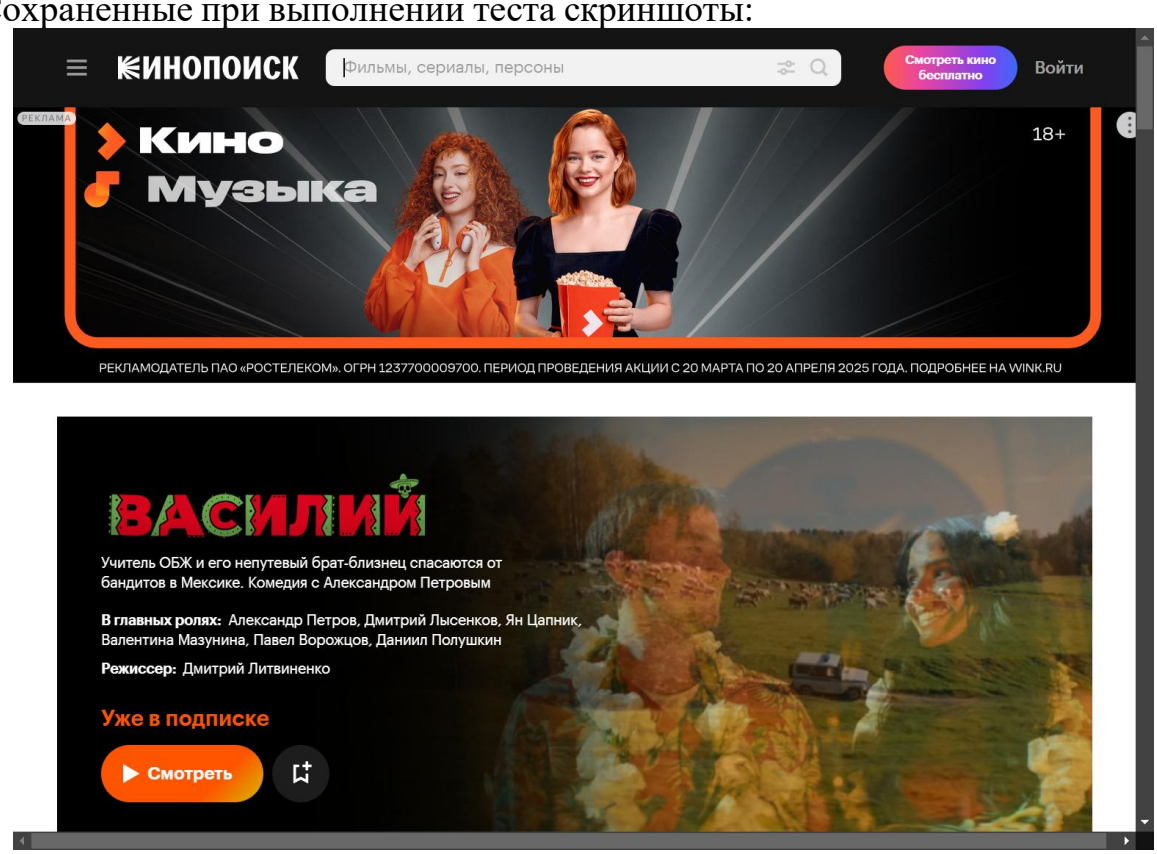

Рисунок 5 – Главная страница сайта

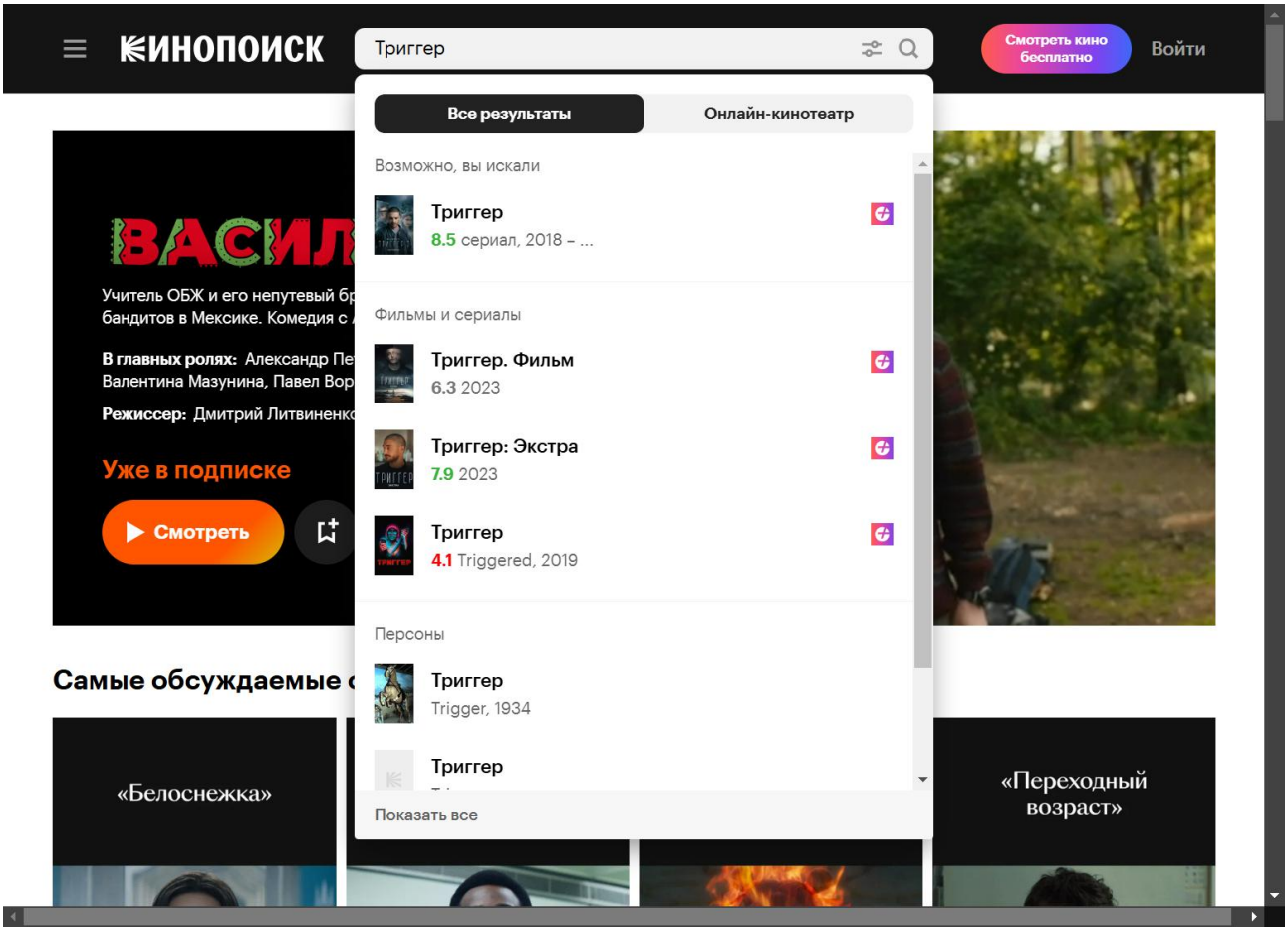

Рисунок 6 – Поисковой запрос

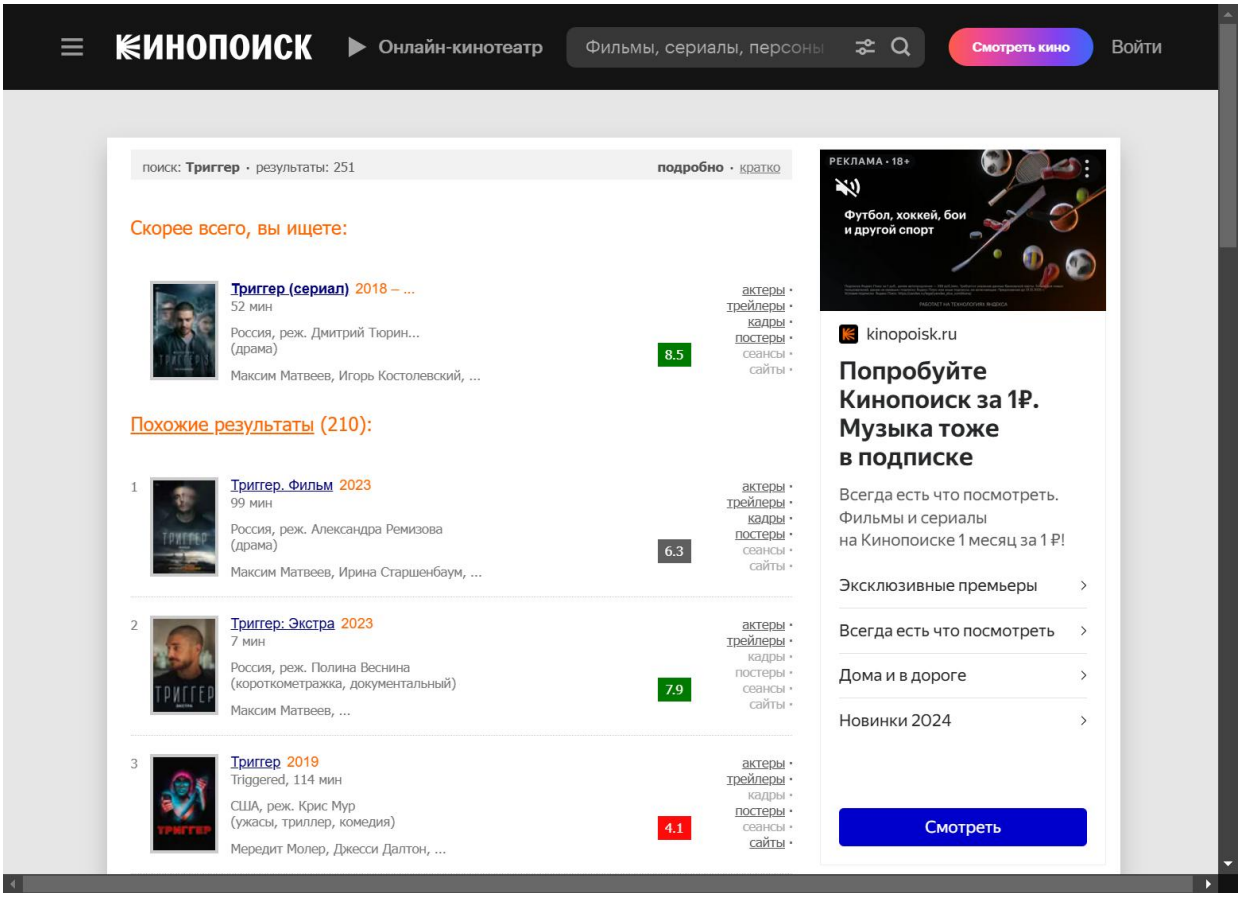

Рисунок 7 – Результат поиска

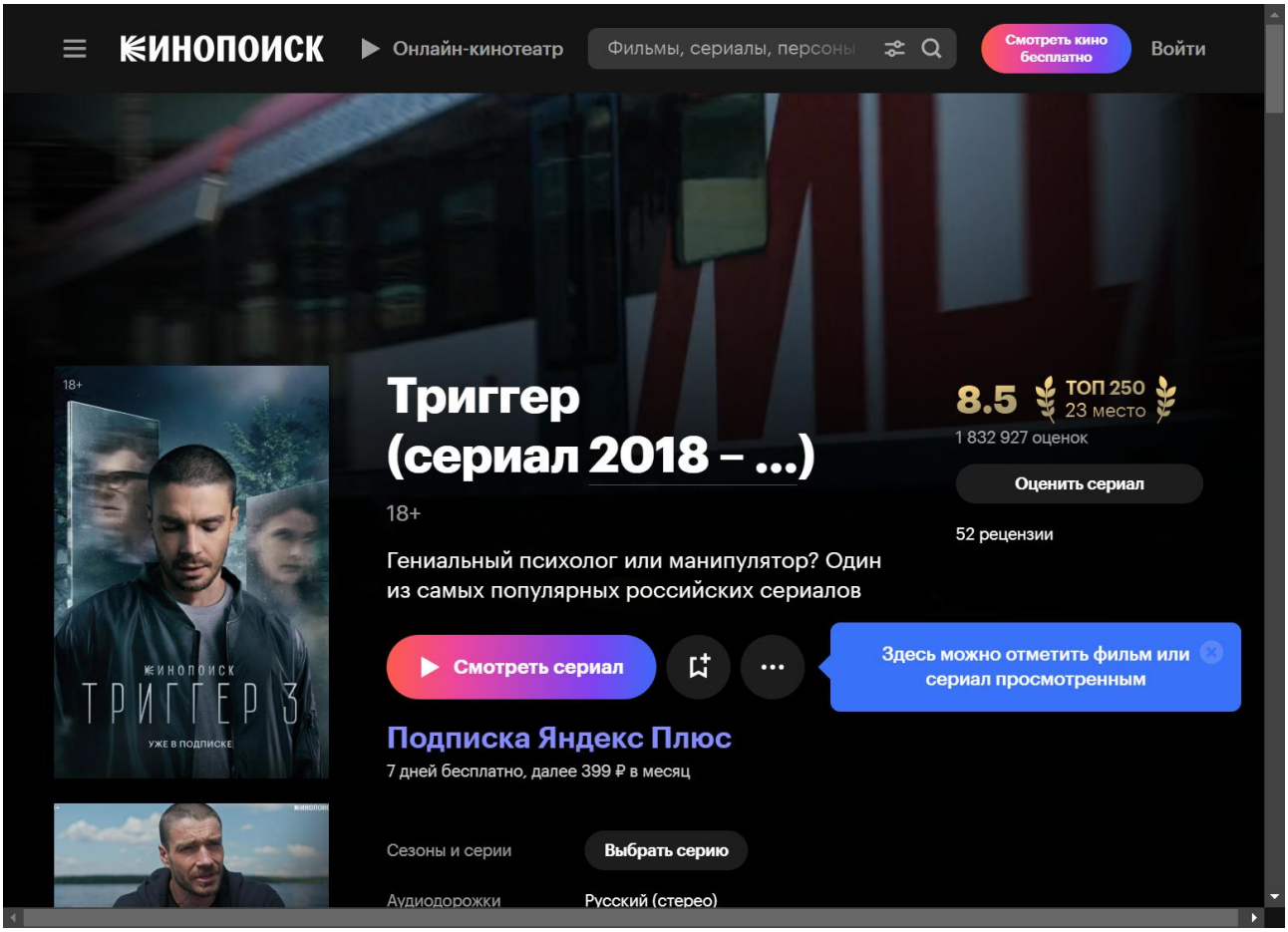

Рисунок 8 – Страница фильма

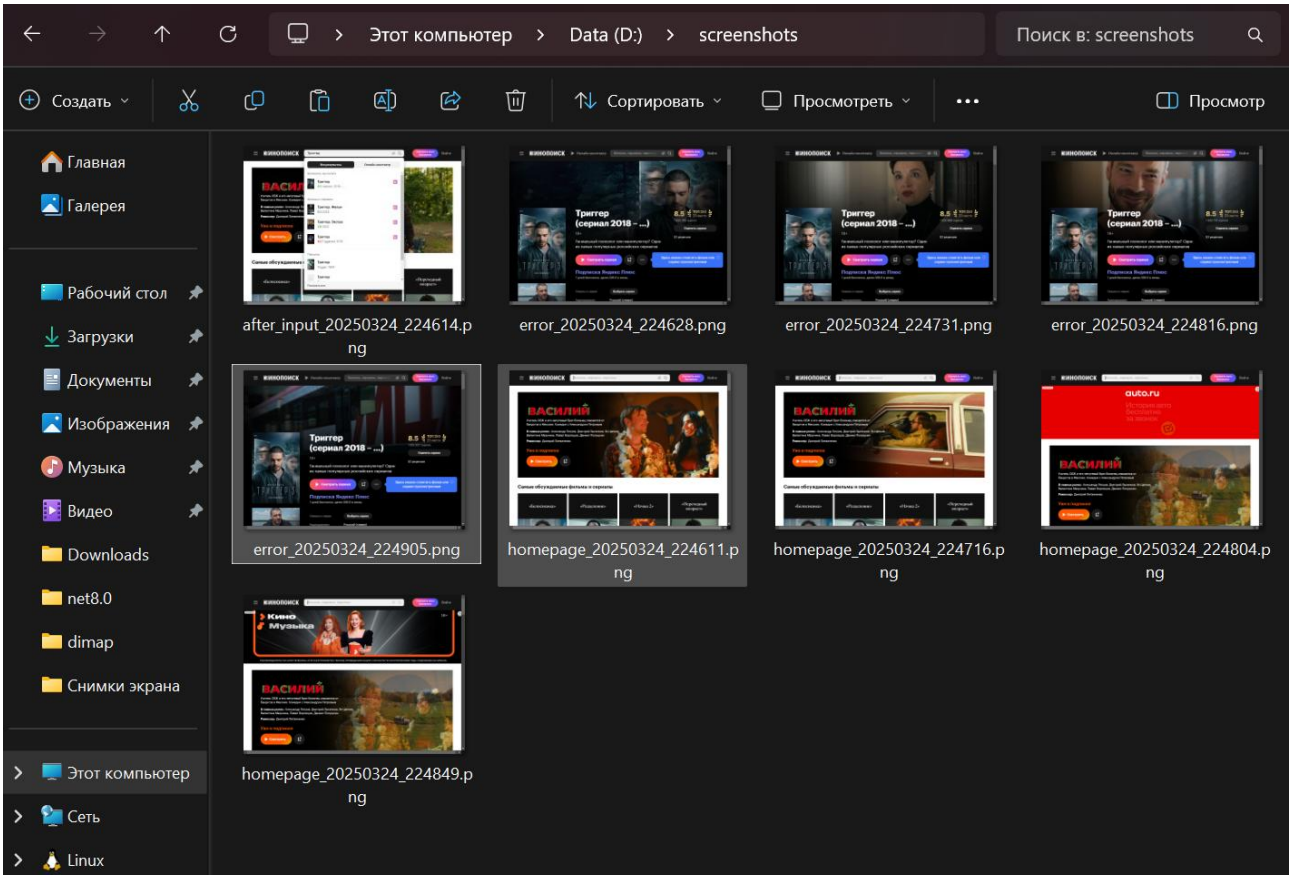

Рисунок 9 – Сохраненные скриншоты теста в папке screenshots

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
```

```python
def create_screenshot_folder(folder_name="screenshots"):
    if not os.path.exists(folder_name):
        os.makedirs(folder_name)
    return folder_name

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}")

def test_invalid_phone_login():
    screenshot_folder = create_screenshot_folder()
    browser = webdriver.Chrome()
    browser.maximize_window()

    try:
        browser.get("https://www.kinopoisk.ru/")
        time.sleep(5)
        input("Пройди капчу вручную и нажми Enter...")
        save_screenshot(browser, "homepage", screenshot_folder)

        login_button = WebDriverWait(browser, 10).until(
            EC.element_to_be_clickable((By.XPATH, "//button[contains(text(),
'Войти') or contains(text(), 'Вход')]")
        ))
        login_button.click()
        save_screenshot(browser, "login_form", screenshot_folder)

        phone_input = WebDriverWait(browser, 10).until(
            EC.presence_of_element_located((By.ID, "passp-field-phone"))
        )
        phone_input.click()  # Клик по полю для активации
        phone_input.clear()  # Очистка поля перед вводом
        phone_input.send_keys("123456")
        save_screenshot(browser, "invalid_phone", screenshot_folder)

        continue_button = WebDriverWait(browser, 10).until(
            EC.element_to_be_clickable((By.XPATH, "//button[contains(text(),
'Продолжить')]")
        ))
        continue_button.click()
        save_screenshot(browser, "after_submit", screenshot_folder)

        error_message = WebDriverWait(browser, 10).until(
            EC.presence_of_element_located((By.XPATH, "//div[contains(@class,
'error') or contains(text(), 'Некорректный номер телефона')]")
        ))
        if error_message:
            save_screenshot(browser, "error_message", screenshot_folder)
        else:
            print(" Ошибка не была найдена!")
```

```
    except Exception as e:
        print(" Ошибка в тесте:", e)
        save_screenshot(browser, "error", screenshot_folder)

    finally:
        browser.quit()
        print(" Браузер закрыт.")

if __name__ == "__main__":
    test_invalid_phone_login()
```

Сохраненные при выполнении теста скриншоты:

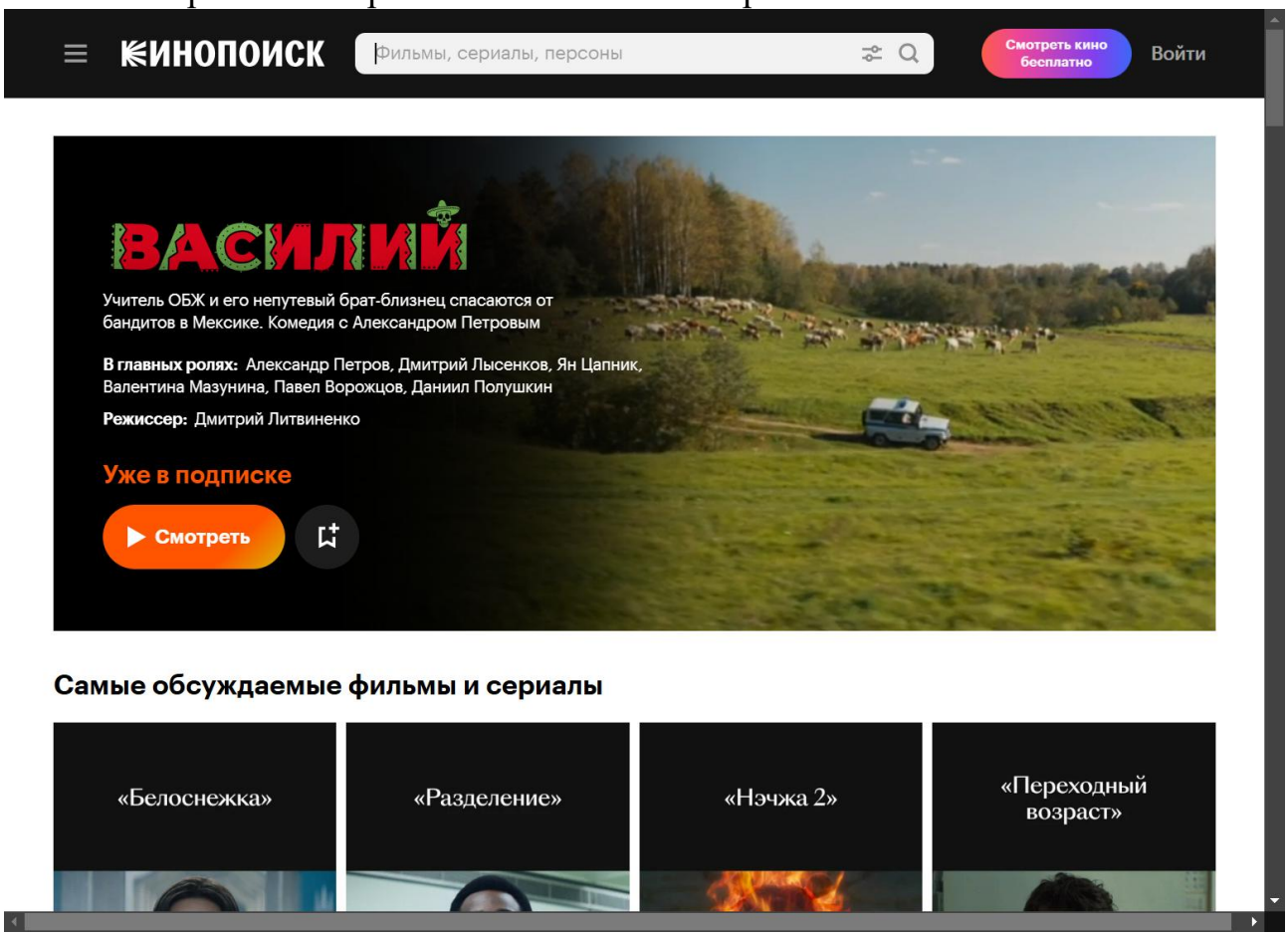

Рисунок 11 – Главная страница

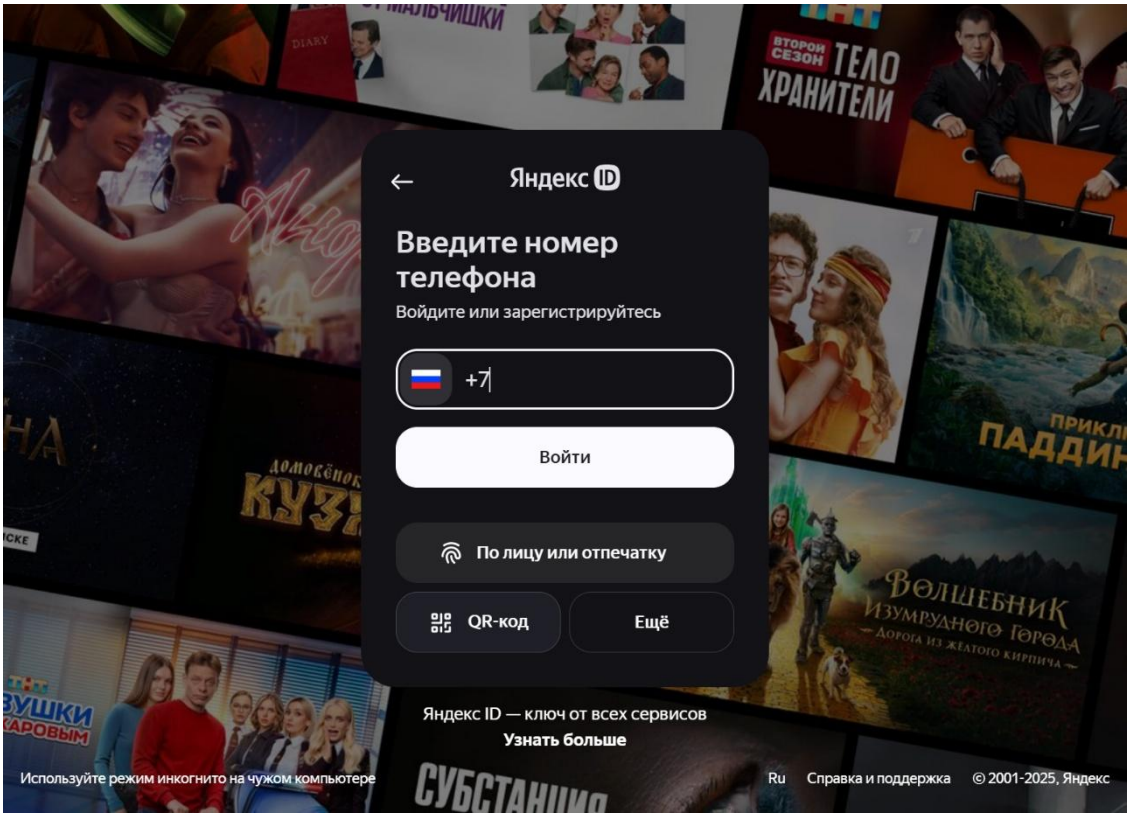

Рисунок 12 – Окно, после нажатия кнопки «Войти»

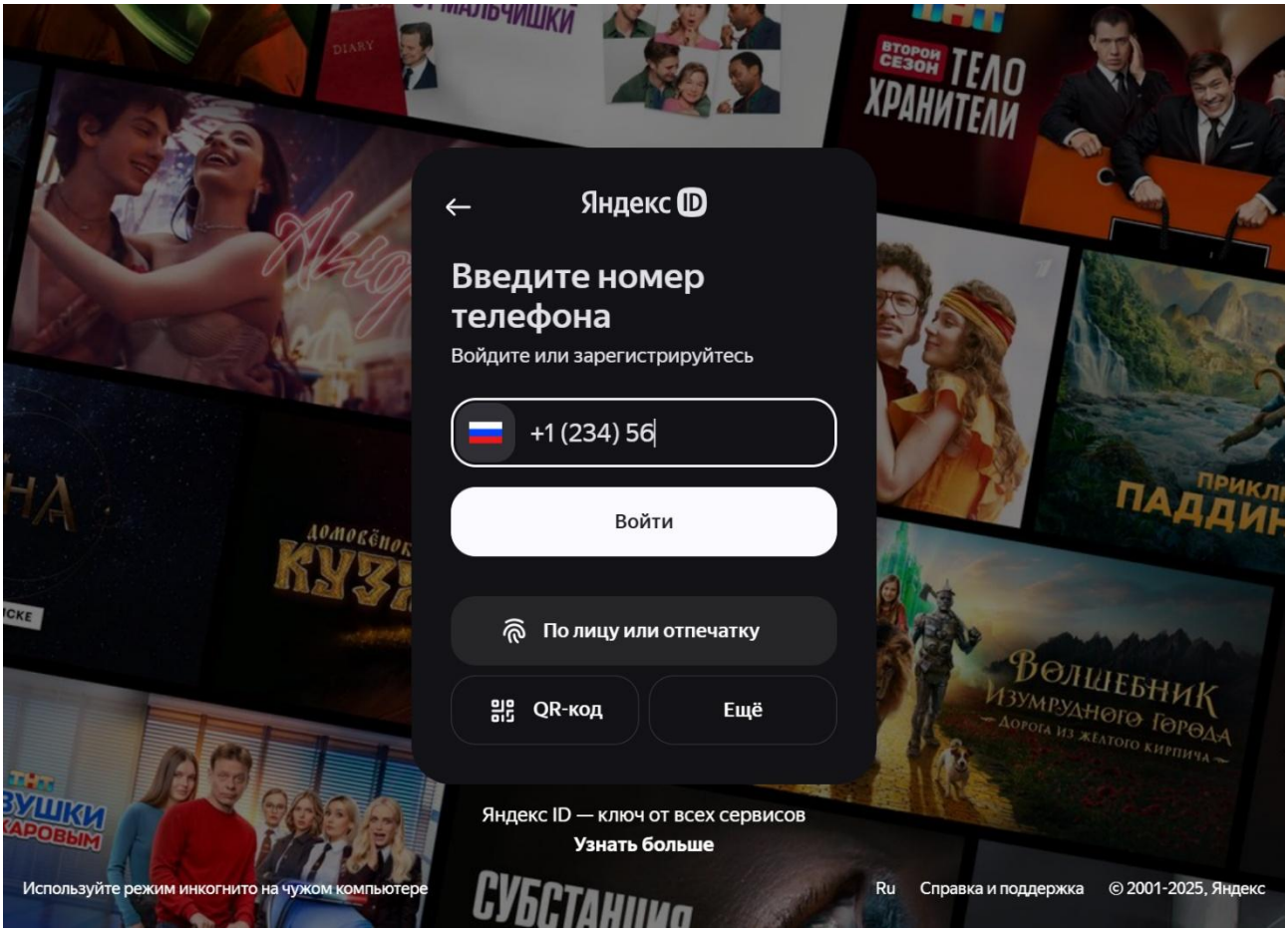

Рисунок 13 – Ввод некорректного номера

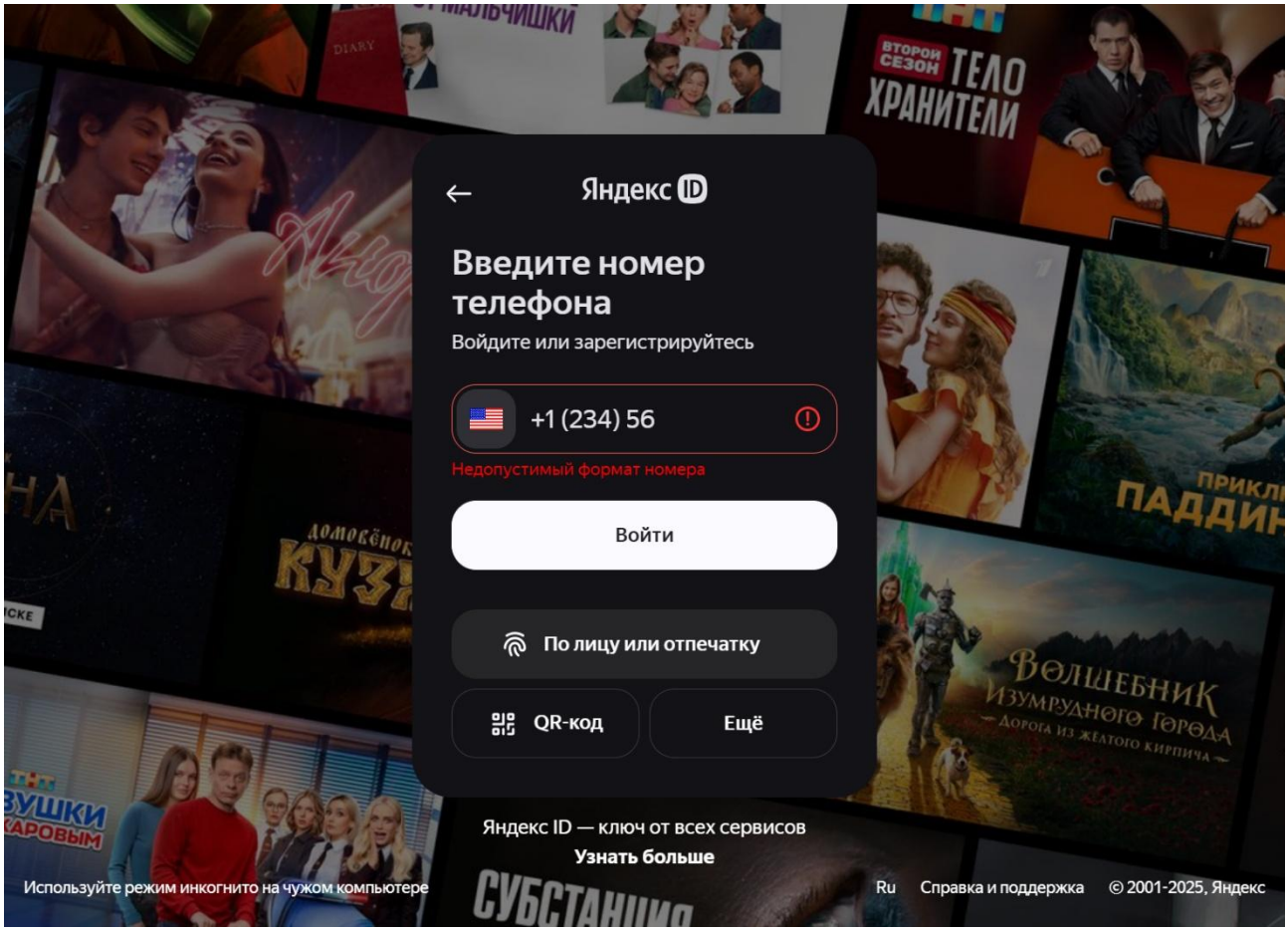

Рисунок 14 – Результат, после нажатия кнопки «Получить код»

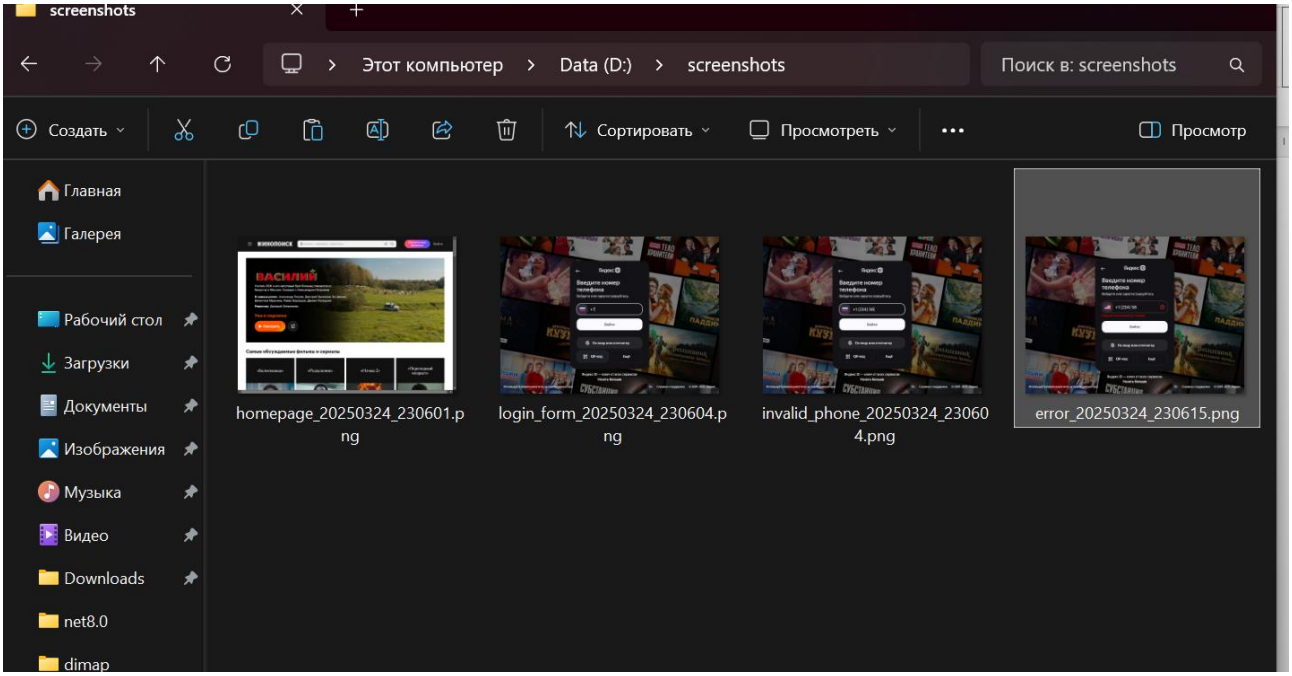

Рисунок 16 – Сохраненные скриншоты теста в папке screenshots