

Lesson 07 Demo 08

Optimizing Test Execution Using GenAI

Objective: To optimize test execution by automating the generation and interpretation of test scripts for improving the functionality and user experience of online library search features

Tools required: ChatGPT 4

Prerequisites: None

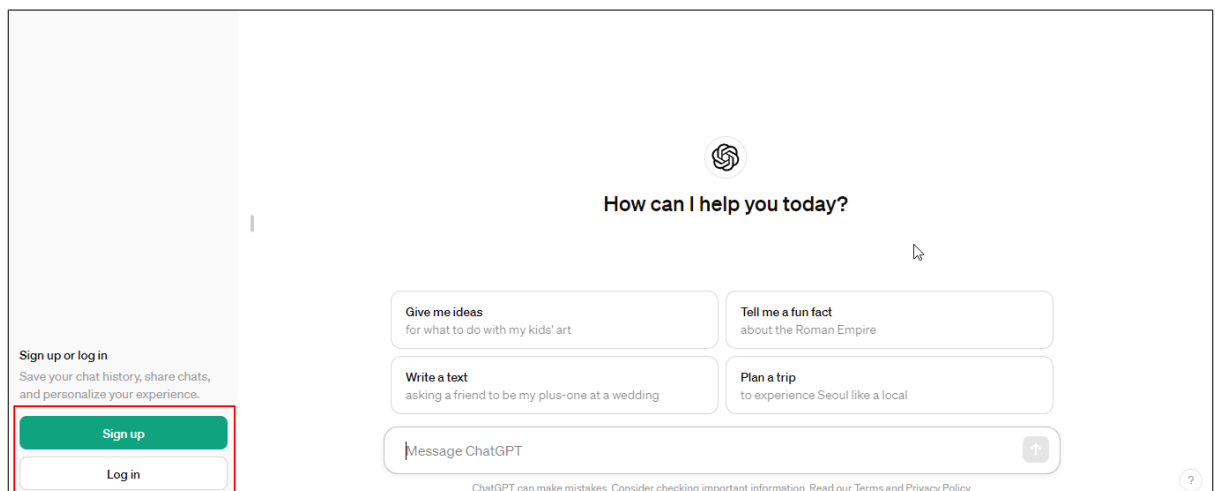
Steps to be followed:

1. Generate test scripts for key functionalities using ChatGPT
2. Interpret test results using ChatGPT

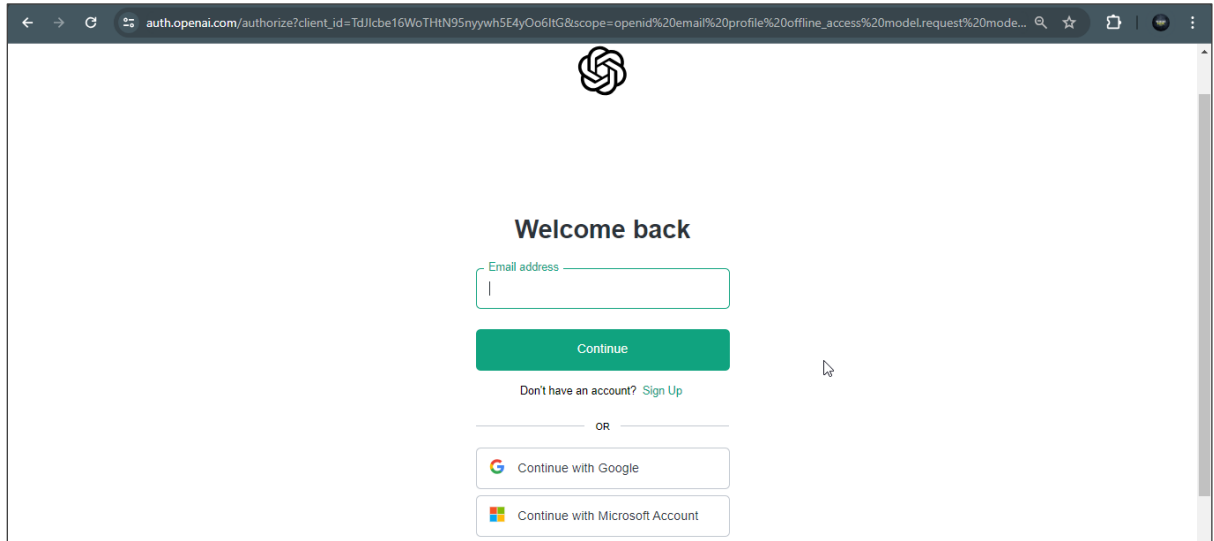
Note: Please be advised that ChatGPT, as an artificial intelligence tool, can produce varied outputs even when presented with similar prompts.

Step 1: Generate test scripts for key functionalities using ChatGPT

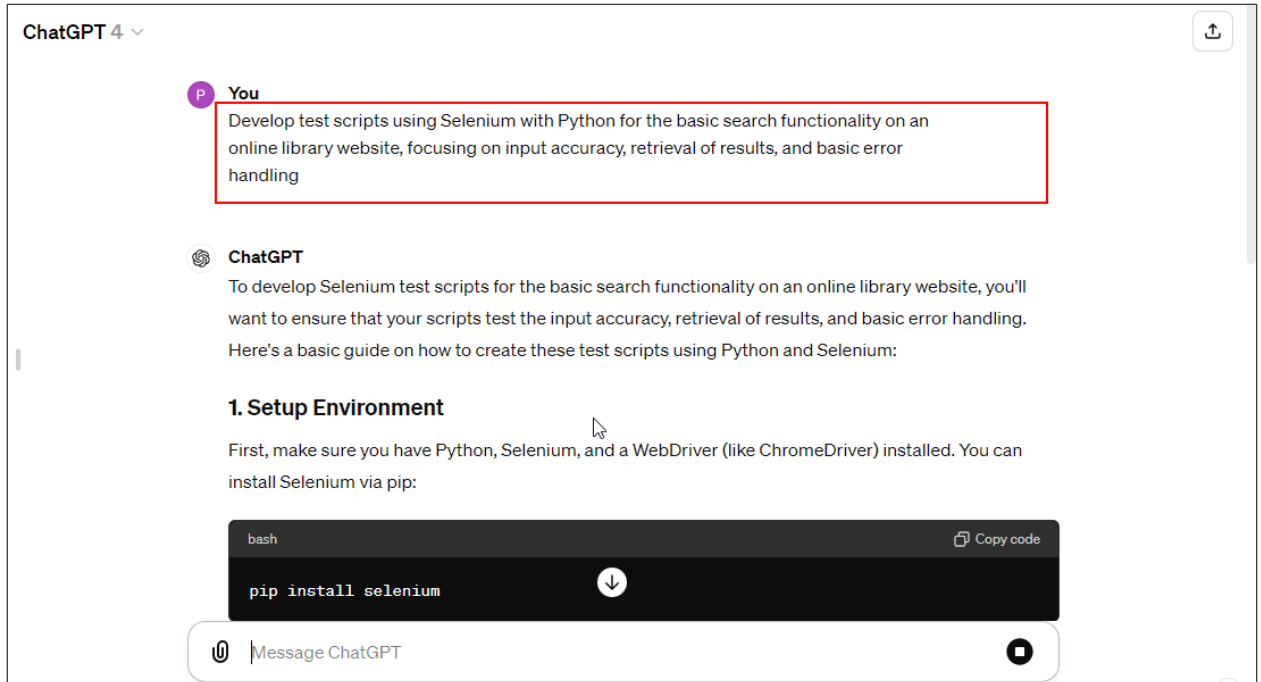
1.1 Browse to the <https://chat.openai.com> website and log in to your account



Note: Sign up if you do not have an account



- 1.2 Use the following prompt to generate test scripts, as shown in the screenshot below:
Develop test scripts using Selenium with Python for the basic search functionality on an online library website, focusing on input accuracy, retrieval of results, and basic error handling



ChatGPT 4

```
python
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
import time

# Setup WebDriver
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))

# Open the target URL
driver.get("https://example-library.com") # Change this URL to the online library's

# Locate the search box (change 'searchBox' to the actual element ID or name on the w
search_box = driver.find_element(By.NAME, "searchBox")

# Test 1: Input accuracy - Enter a kn book title and press ENTER
search_box.send_keys("To Kill a Mockingbird")
```

Copy code

1.3 Use the prompt to generate advanced Selenium search scripts with multifunctionality, as shown in the screenshot below:

Create advanced Selenium test scripts for filtering and sorting mechanisms in the search functionality of an online library website, verifying accuracy and response time under different network conditions

ChatGPT 4

You

Create advanced Selenium test scripts for filtering and sorting mechanisms in the search functionality of an online library website, verifying accuracy and response time under different network conditions

ChatGPT

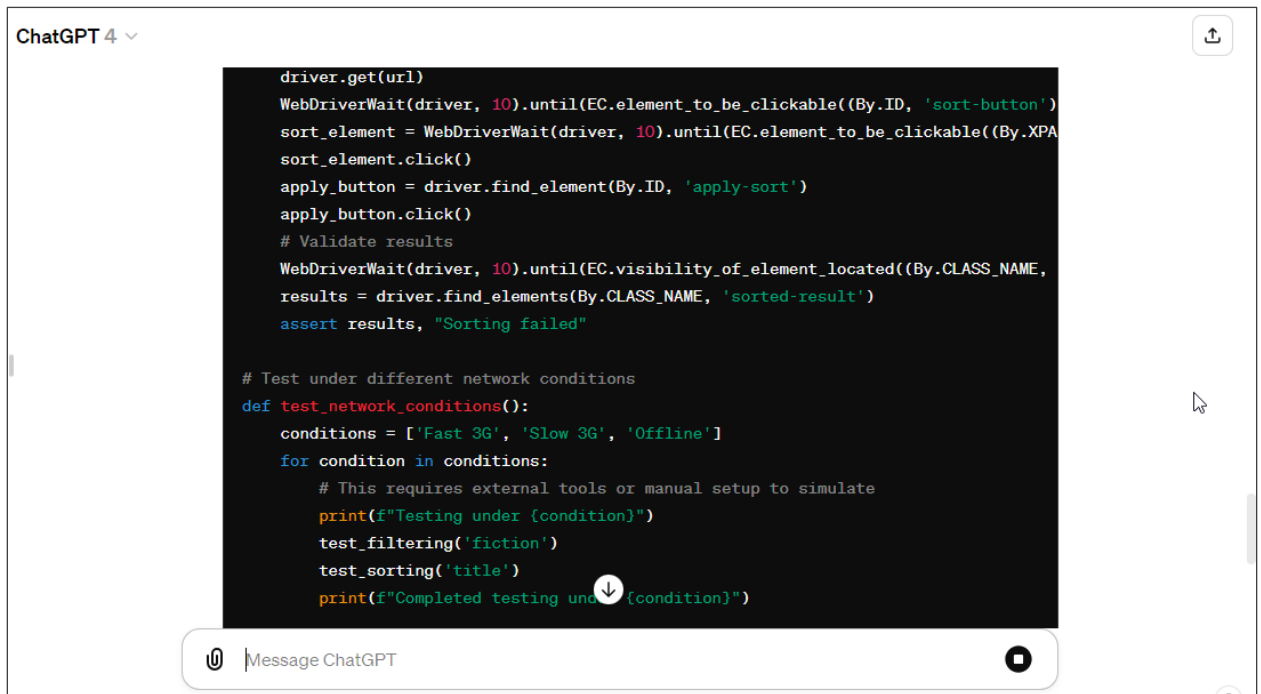
To create advanced Selenium test scripts for filtering and sorting mechanisms on an online library website, you will focus on verifying the accuracy of the search results and the response time under different network conditions. This involves complex Selenium interactions and possibly integrating with other tools to simulate network conditions.

1. Setup Environment

First, ensure that you have Python, Selenium, and a WebDriver installed. You will also need `webdriver_manager` to handle driver installations and `requests` to simulate network conditions if not using an external tool.

bash

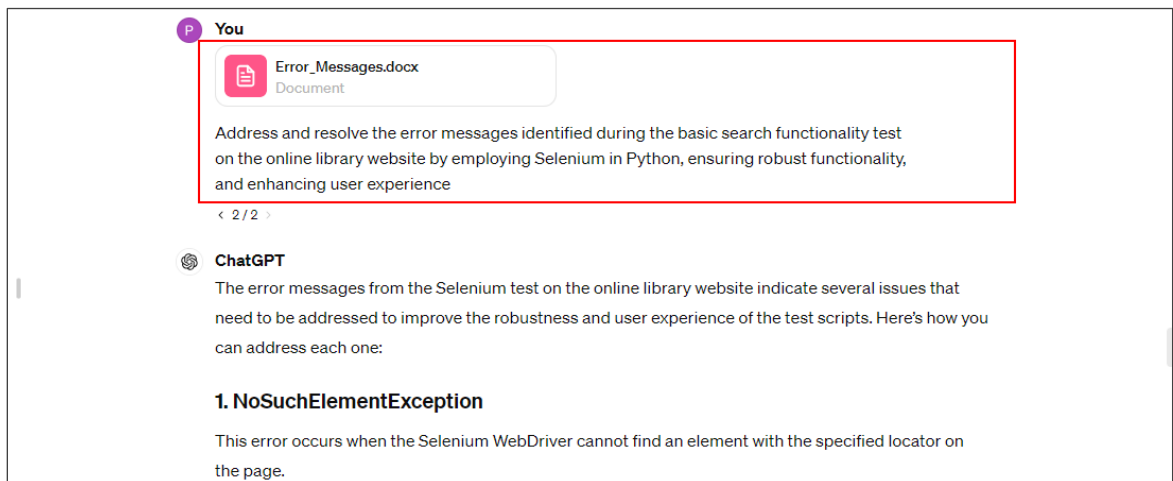
Copy code



Step 2: Interpret test results using ChatGPT

2.1 Upload the **Error_Messages.docx** file and use the following prompt to analyze and resolve errors, as shown in the screenshot below:

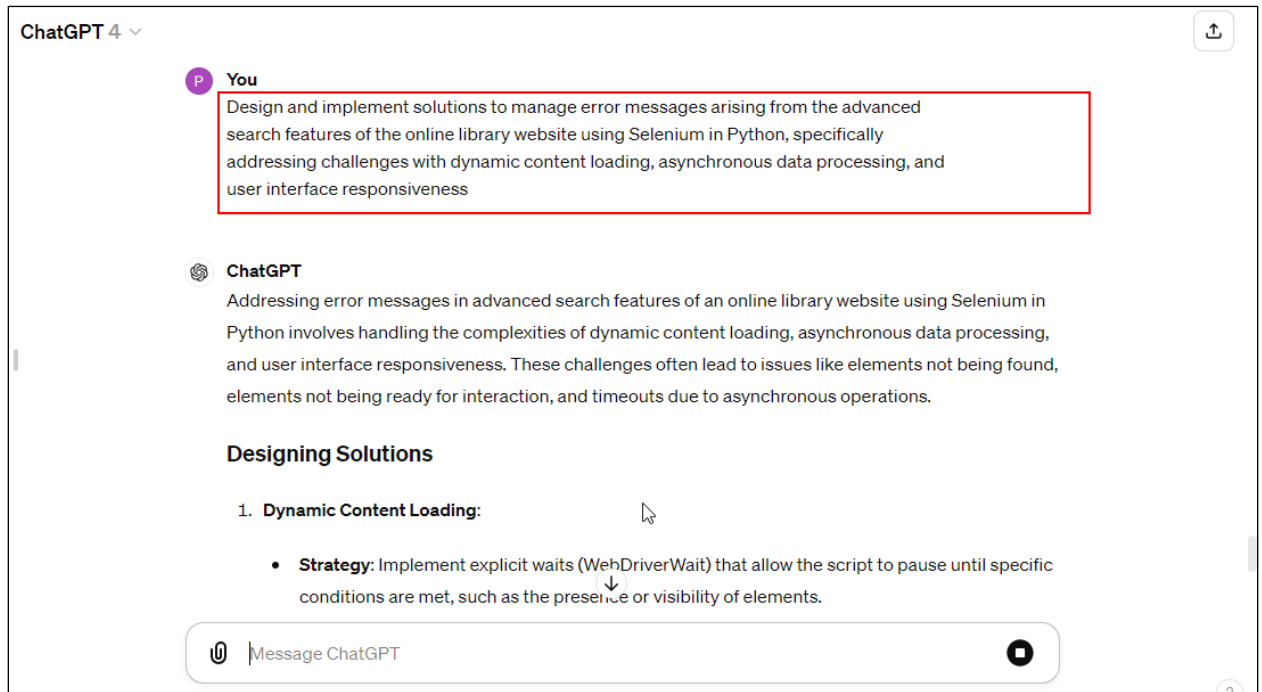
Address and resolve the error messages identified during the basic search functionality test on the online library website by employing Selenium in Python, ensuring robust functionality, and enhancing user experience



Note: Download the **Error_Messages.docx** file from the LMS course resources

2.2 Use the following prompt to design solutions for error management, as shown in the screenshot below:

Design and implement solutions to manage error messages arising from the advanced search features of the online library website using Selenium in Python, specifically addressing challenges with dynamic content loading, asynchronous data processing, and user interface responsiveness



By following these steps, you have successfully optimized test execution by using ChatGPT to automate the generation and interpretation of test scripts to enhance both functionality and user experience for online library search features.