1. Using Selenium WebDriver in Python, to Trigger a JavaScript prompt alert then Send the text "Test User" to the alert then Accept the alert finally Print the alert message before entering text. Ans: selenium\_alert.py:

|  |
| --- |
| from selenium import webdriver  from selenium.webdriver.common.alert import Alert  from selenium.webdriver.common.by import By  from selenium.webdriver.chrome.service import Service  from webdriver\_manager.chrome import ChromeDriverManager  import time  import os  # Set up the WebDriver  driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))  # HTML content with JavaScript prompt and alert  html\_content = """  <html>  <body>  <script>  function triggerPrompt() {  var message = prompt("Please enter your name:");  if (message !== null) {  alert("You entered: " + message);  }  }  </script>  <button onclick="triggerPrompt()">Trigger Prompt</button>  </body>  </html>  """  # Save the HTML content to a temporary file  with open("temp.html", "w") as f:  f.write(html\_content)  # Open the local HTML file in the browser  driver.get("file://" + os.path.abspath("temp.html"))  # Click the button to trigger the prompt  driver.find\_element(By.TAG\_NAME, "button").click()  time.sleep(1) # Wait for prompt alert  # Handle the prompt alert  alert = Alert(driver)  print(f"Prompt says: {alert.text}")  alert.send\_keys("Test User")  alert.accept()  time.sleep(1) # Wait for confirmation alert  # Handle the confirmation alert  alert = Alert(driver)  print(f"Final alert says: {alert.text}")  alert.accept()  # Close the browser  driver.quit() |

2. Write a Selenium script to Wait explicitly until an alert is present (up to 10 seconds).Then Print the text of the alert. Accept it.

 Note: Use below given link to perform actions in Task 1 and 2.

"https://www.selenium.dev/documentation/webdriver/interactions/alerts/

|  |
| --- |
| from selenium import webdriver  from selenium.webdriver.common.by import By  from selenium.webdriver.chrome.service import Service  from selenium.webdriver.support.ui import WebDriverWait  from selenium.webdriver.support import expected\_conditions as EC  from selenium.webdriver.common.alert import Alert  from webdriver\_manager.chrome import ChromeDriverManager  # Set up the WebDriver  driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))  # Navigate to the provided URL  driver.get("https://www.selenium.dev/documentation/webdriver/interactions/alerts/")  # Find and click the button that triggers a prompt alert  driver.find\_element(By.LINK\_TEXT, "See a sample prompt").click()  # Explicitly wait for the alert to be present (up to 10 seconds)  wait = WebDriverWait(driver, 10)  alert = wait.until(EC.alert\_is\_present())  # Get the alert text and print it  alert\_text = alert.text  print(f"Alert message: {alert\_text}")  # Accept the alert  alert.accept()  # Close the browser  driver.quit() |

3. Research and implement a Fluent Wait strategy using Selenium and WebDriverWait with polling

intervals to wait for a changing element text.

|  |
| --- |
| from selenium import webdriver from selenium.webdriver.common.by import By from selenium.webdriver.chrome.service import Service from selenium.webdriver.support.ui import WebDriverWait from selenium.webdriver.support import expected\_conditions as EC from webdriver\_manager.chrome import ChromeDriverManager import time  # Set up WebDriver driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))  try:  # Navigate to a reliable site  driver.get("https://the-internet.herokuapp.com/dynamic\_loading/2")   # Click the 'Start' button to trigger loading  start\_button = WebDriverWait(driver, 10).until(  EC.element\_to\_be\_clickable((By.CSS\_SELECTOR, "#start button"))  )  start\_button.click()   # Wait for the loading to finish and text to appear  finish\_element = WebDriverWait(driver, 15).until(  EC.visibility\_of\_element\_located((By.ID, "finish"))  )   # Print the final text  final\_text = finish\_element.text.strip()  print(f"✅ Final loaded text: {final\_text}")  finally:  time.sleep(2)  driver.quit() |

4. Write a Python Selenium script that sets an implicit wait, opens the Selenium dynamic elements

demo page ("https://www.selenium.dev/selenium/web/dynamic.html”), clicks the "adder" button

|  |
| --- |
| from selenium import webdriver  from selenium.webdriver.common.by import By  from selenium.webdriver.chrome.service import Service  from webdriver\_manager.chrome import ChromeDriverManager  import time  # Set up the WebDriver  driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))  # Set implicit wait to 10 seconds  driver.implicitly\_wait(10)  try:  # Navigate to the Selenium dynamic elements demo page  driver.get("https://www.selenium.dev/selenium/web/dynamic.html")  # Find and click the "adder" button  adder\_button = driver.find\_element(By.ID, "adder")  adder\_button.click()  finally:  # Close the browser  time.sleep(2) # Brief pause to observe result  driver.quit() |