Misr Digital innovation bank

Under supervised/ Test Team lead. Ahmed Yossri

Amazon scenario Task

Name: Mohamed Osama

Phone :01277989285

Gmail. mohamedosama12546@gmail.com

Faculty: faculty of computer and artificial intelligence at Cairo university

link task Github:
https://github.com/Mohamedosama-dev/Amazon-scenario-Task-

MISR DIGITAL INNOVATION

Deception of task:

Step-by-Step Description

1. **Open Amazon Egypt and Login**
   - Launch the browser and navigate to `https://www.amazon.eg/`.
   - Locate and interact with the login fields (email and password) to log into your account.
2. **Open the "All" Menu**
   - Identify the "All" menu button on the left side of the homepage.
   - Click on the "All" menu to reveal the categories.
3. **Select "Video Games"**
   - From the dropdown menu, find and click on "Video Games."
   - Then, choose the "All Video Games" option.
4. **Apply Filters**
   - In the filter menu on the left side, check the box for “Free Shipping.”
   - Additionally, check the box for the condition “New” to filter for new items only.
5. **Sort Products by Price**
   - On the right side of the page, find the sort menu.
   - Select the option to sort by price in descending order (high to low).
6. **Add Products to Cart**
   - Check for products priced below 15,000 EGP.
   - If products are found, add each product below this price to the cart.
   - If no products are found below this price, navigate to the next page and repeat the process until all pages are checked.
7. **Verify All Products are in Cart**
   - Navigate to the cart to ensure all selected products have been successfully added.
8. **Add Shipping Address and Choose Payment Method**
   - Proceed to checkout.
   - Select "Cash on Delivery" as the payment method.
9. **Verify Total Amount**

Explanation of Task:

1-

```
// Setup WebDriver (Firefox driver)
WebDriverManager.firefoxdriver().setup();
WebDriver driver = new FirefoxDriver();
```

setup web driver firefox

2-

```
// Start of execution
System.out.println("Starting the Amazon script...");

// Open Amazon Egypt website
driver.get("https://www.amazon.eg/");
driver.manage().window().maximize();
System.out.println("Opened Amazon Egypt website.");

WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(100));
```

The code opens the Amazon Egypt website and maximizes the browser window for better visibility. It logs a message to the console confirming that the site has been successfully accessed. Additionally, a `WebDriverWait` is created with a 100-second timeout to manage delays in loading page elements before interaction.

3-

```
// Login
login(driver, wait);
```

```
private static void login(WebDriver driver, WebDriverWait wait) {  1 usage
    System.out.println("Waiting for 'Sign In' button...");
    WebElement signIn = wait.until(ExpectedConditions.elementToBeClickable(By.id("nav-link-accountList-nav-line-1")));
    signIn.click();
    System.out.println("Clicked 'Sign In' button.");
    // 1. Login
    System.out.println("Waiting for email input...");
    WebElement email = wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("ap_email")));
    email.sendKeys( ...keysToSend: "01277989285");
    driver.findElement(By.id("continue")).click();
    System.out.println("Entered email and clicked 'Continue'.");
    System.out.println("Waiting for password input...");
    WebElement password = wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("ap_password")));
    password.sendKeys( ...keysToSend: "Mohamed123@@@");
    driver.findElement(By.id("signInSubmit")).click();
    System.out.println("Entered password and clicked 'Sign In'.");
}
```

The `login` method uses Selenium WebDriver to automate the Amazon sign-in process by waiting for and clicking the "Sign In" button. It enters the email (`01277989285`) and password (`Mohamed123@@@`) after ensuring the corresponding input fields are visible

4-

```
WebElement allMenu = wait.until(ExpectedConditions.elementToBeClickable(By.id("nav-hamburger-menu")));
allMenu.click();
System.out.println("Clicked 'All' menu button.");

// 3. Click on "See All"
WebElement seeAll = wait.until(ExpectedConditions.elementToBeClickable(By.xpath( xpathExpression: "//i[@class='nav-sprite hmenu-arrow-more']"
seeAll.click();
System.out.println("Clicked 'See All'.");

// 4. Click on "Video Games"
WebElement videoGames = wait.until(ExpectedConditions.elementToBeClickable(By.xpath( xpathExpression: "//div[contains(text(),'Video Games')]"
videoGames.click();
System.out.println("Clicked 'Video Games'.");
```

the code opens the "All" menu on Amazon by clicking the hamburger menu button after ensuring it is clickable. It then clicks the "See All" option and navigates to the "Video Games" category, confirming each action with console logs.

5-

```
// 5. Scroll to "All Video Games" and click
WebElement allVideoGames = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath( xpathExpression: "//a[contains(text(),'All V
try {
    allVideoGames.click();
    System.out.println("Clicked 'All Video Games'");
} catch (Exception e) {
    ((JavascriptExecutor) driver).executeScript( script: "arguments[0].click();", allVideoGames);
    System.out.println("Clicked 'All Video Games' ");
}
// Wait for URL to change to confirm navigation
wait.until(ExpectedConditions.urlContains( fraction: "nav_em_vg_all"));
System.out.println("URL changed, page has loaded successfully.");

// Check if 'Video Games' section is visible
WebElement videoGamesTitle = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath( xpathExpression: "//*[contains(text(),'Vid
System.out.println("Successfully navigated to 'Video Games' section.");
```

The code scrolls to the "All Video Games" link and attempts to click it, using JavaScript execution if the standard click fails, logging the action either way. It then waits for the URL to change, confirming successful navigation to the relevant section. Finally, it checks for the visibility of the "Video Games" title to ensure the page has loaded correctly, logging a confirmation message.

6-

```
// 6. Click on "Free Shipping" filter
applyFilter(wait, driver, filterName: "Free Shipping");
```

```
private static void applyFilter(WebDriverWait wait, WebDriver driver, String filterName) {  2 usages
    try {
        WebElement filter = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath( xpathExpression: "//span[contains(text(), '" + filterN
        filter.click();
        System.out.println("Clicked '" + filterName + "' filter.");
    } catch (Exception e) {
        System.out.println("Failed to find '" + filterName + "' filter. Attempting to click using JavaScript...");
        WebElement filter = driver.findElement(By.xpath( xpathExpression: "//span[contains(text(), '" + filterName + "')]"));
        ((JavascriptExecutor) driver).executeScript( script: "arguments[0].scrollIntoView(true);", filter);
        ((JavascriptExecutor) driver).executeScript( script: "arguments[0].click();", filter);
        System.out.println("Clicked '" + filterName + "' filter using JavaScript.");
    }
}
```

The `applyFilter` method attempts to click a specified filter on the Amazon page by first waiting for its visibility and clicking it, logging the action. If the filter is not found, it catches the exception, scrolls the filter into view, and then clicks .

7-

```
// 8. Sort by "Price: High to Low"
WebElement sortMenu = wait.until(ExpectedConditions.elementToBeClickable(By.xpath( xpathExpression: "//span[contains(text(), 'Sort by')]")));
sortMenu.click();
WebElement highToLow = wait.until(ExpectedConditions.elementToBeClickable(By.xpath( xpathExpression: "//a[contains(text(), 'Price: High to Low')]")));
highToLow.click();
System.out.println("Clicked 'Price: High to Low' to sort.");
```

The code waits for the sort menu to be clickable and then clicks it to reveal sorting options. It subsequently waits for the "Price: High to Low" option to become clickable and clicks it, logging the action to confirm the sorting choice.

8-

```
// Loop through all pages, adding products to the cart if they are below 15k EGP
boolean hasNextPage = true;
while (hasNextPage) {
    System.out.println("Processing current page...");

    List<WebElement> products = driver.findElements(By.xpath( xpathExpression: "//div[contains(@class, 's-main-slot')]/div[@data-asin]"));
    boolean addedToCart = false;

    for (WebElement product : products) {
        try {
            String priceText = getProductPrice(product);
            if (priceText != null) {
                double price = parsePrice(priceText);
                if (price < 15000) {
                    System.out.println("Adding product with price " + price + " EGP to cart...");
                    addItemToCart(driver, product);
                    addedToCart = true;
                }
            }
```

```
        if (addedToCart) {
            System.out.println("Products have been added to cart.");
        } else {
            System.out.println("No products below 15k EGP found on this page.");
        }

        // Handle pagination to go to the next page
        try {
            WebElement nextPageButton = wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath( xpathExpression: "//a[contains(@class,
            if (nextPageButton.isEnabled()) {
                nextPageButton.click();
                System.out.println("Clicked next page button.");
                Thread.sleep( millis: 3000); // Wait for the page to load
            } else {
                System.out.println("Next page button is disabled. Exiting.");
                hasNextPage = false; // Exit if the button is disabled
            }
```

The code sorts the products by price in descending order and then enters a loop to process all available pages, adding products priced below 15,000 EGP to the cart. It retrieves and checks the price of each product, logging whether items were added or if none were found on the current page. After processing the products, it attempts to navigate to the next page by clicking the "Next" button, handling potential issues with pagination through exception handling and retries.

9-

```
private static void proceedToCheckout(WebDriver driver, WebDriverWait wait) throws InterruptedException {  1 usage

    // Step 1: Click "Go to basket"
    WebElement goToBasketButton = wait.until(ExpectedConditions.elementToBeClickable(By.id("nav-cart")));
    goToBasketButton.click();
    System.out.println("Clicked 'Go to basket'.");
    //step 2 get priceElement
    //check price for all products
    WebElement priceElement =wait.until(ExpectedConditions.elementToBeClickable(By.xpath( xpathExpression: "(//span[@class='a-size-medium a-color
    String priceText = priceElement.getText();
    System.out.println("Extracted Price: " + priceText);
```

```
// Step 3: Click "Proceed to Buy"
WebElement proceedToBuyButton = wait.until(ExpectedConditions.elementToBeClickable(By.xpath( xpathExpression: "//input[@name='proceedToRetailCheck
proceedToBuyButton.click();
System.out.println("Clicked 'Proceed to Buy'.");
// Step 4: Click "Change" for shipping address
WebElement shippingAddressButton = wait.until(ExpectedConditions.elementToBeClickable(By.xpath( xpathExpression: "//a[@id='addressChangeLinkId' a
Thread.sleep( millis: 4000);  // Ensure page has loaded
((JavascriptExecutor) driver).executeScript( script: "arguments[0].scrollIntoView(true);", shippingAddressButton);
((JavascriptExecutor) driver).executeScript( script: "arguments[0].click();", shippingAddressButton);
System.out.println("Clicked 'Change' for shipping address.");
```

```
    // Step 5: Click "Use this address"
    WebElement useThisAddressButton = wait.until(ExpectedConditions.elementToBeClickable(By.xpath( xpathExpression: "//input[@aria-labelledby='order
    Thread.sleep( millis: 4000);  // Ensure page has loaded
    ((JavascriptExecutor) driver).executeScript( script: "arguments[0].click();", useThisAddressButton);
    System.out.println("Clicked 'Use this address'.");
    // Step 6: Select "Cash on Delivery" payment method
    try {
        WebElement cashPaymentOption = wait.until(ExpectedConditions.elementToBeClickable(By.xpath( xpathExpression: "//input[@value='cash-on-delive
        cashPaymentOption.click();
        System.out.println("Selected 'Cash on Delivery' payment method.");
    } catch (Exception e) {
        System.out.println("Error selecting payment method: " + e.getMessage());
    }
}
```

The `proceedToCheckout` method automates the checkout process on Amazon Egypt. It begins by clicking the "Go to basket" to view the cart and extracts the price of the first item. Next, it clicks "Proceed to Buy" to continue to the checkout page. The method updates the shipping address by selecting "Change" and confirming the new address. Finally, it selects the "Cash on Delivery" payment option and handles any errors that may occur during the selection process.

Output:

I make message after each step to insure everything work well

1-

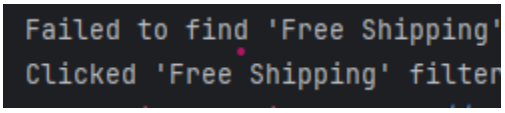
2-

3-

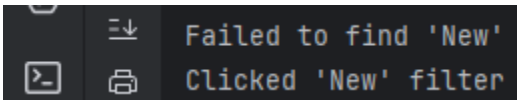
4-

5-

6-



7-



8-

```
Clicked 'Price: High to Low' to sort.
```

9-

```
Adding product with price 13980.0 EGP to cart...
Added product to cart.
Adding product with price 11750.0 EGP to cart...
Added product to cart.
Adding product with price 11200.0 EGP to cart...
Added product to cart.
Adding product with price 10999.0 EGP to cart...
Failed to add product to cart.
Adding product with price 10444.0 EGP to cart...
Failed to add product to cart.
Adding product with price 9999.0 EGP to cart...
Added product to cart.
Adding product with price 9749.0 EGP to cart...
Added product to cart.
Adding product with price 9500.0 EGP to cart...
Added product to cart.
Adding product with price 9499.0 EGP to cart...
Added product to cart.
Adding product with price 9499.0 EGP to cart...
Added product to cart.
Adding product with price 9150.0 EGP to cart...
Added product to cart.
Adding product with price 8999.0 EGP to cart...
Failed to add product to cart.
tal Innovation Task > src > test > java > MDI Bank Amazon
```

10-

```
Clicked 'Go to basket'.
Extracted Price: EGP 4,358.99
Clicked 'Proceed to Buy'.
```

11-

```
Clicked 'Change' for shipping address.
```

12-

```
Clicked 'Change' for shipping address.
```

13-

```
Clicked 'Use this address'.
```

Thank you for your support and guidance throughout this task. I appreciate your leadership and insights that helped me to solve the task
thank You Eng /Ahmed Yossri

Warm regards
Mohamed Osama