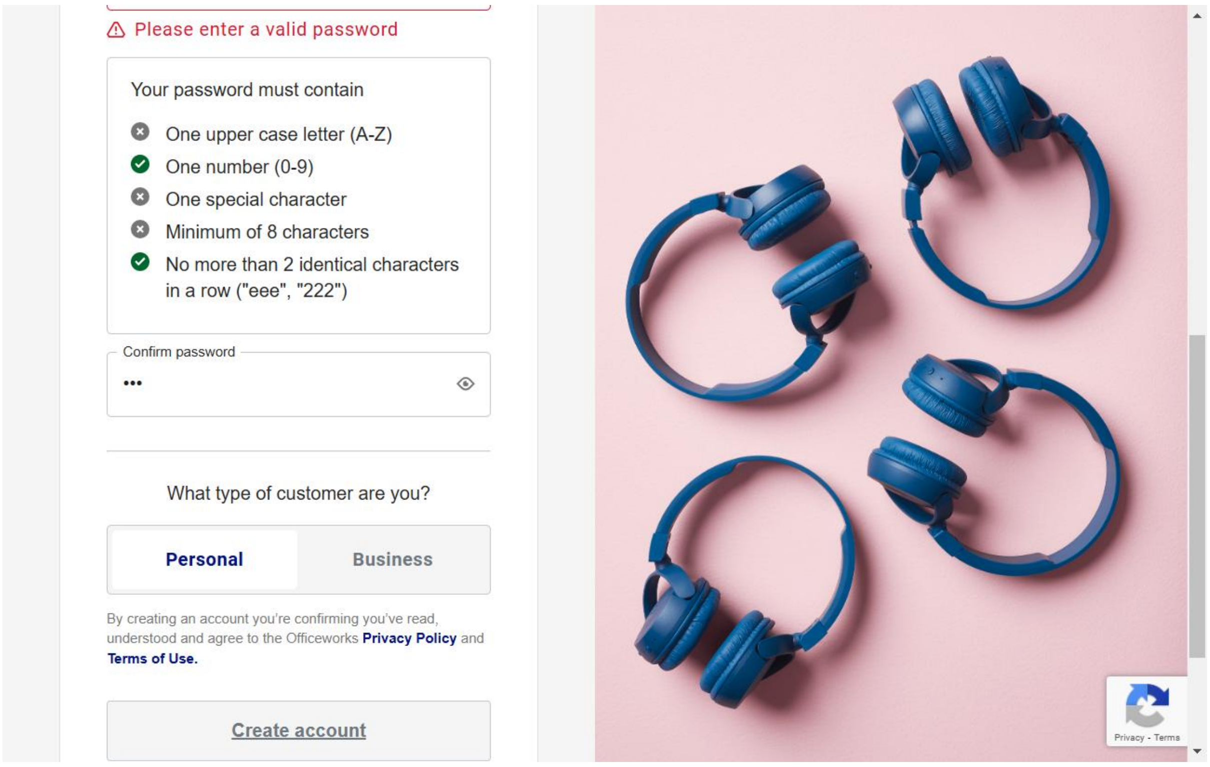
Git repo: https://github.com/TOMINJOSE88/sit707.git

- **The screenshot taken of the officework's filled and submitted registration page(https://www.officeworks.com.au/app/identity/create-account) using Selenium screenshot API. Make sure you fail only the password requirement, so the page does not submit successfully and contains errors. This respects not to spoil an official website.**



Please enter a valid password

Your password must contain

- ✗ One upper case letter (A-Z)
- ✓ One number (0-9)
- ✗ One special character
- ✗ Minimum of 8 characters
- ✓ No more than 2 identical characters in a row ("eee", "222")

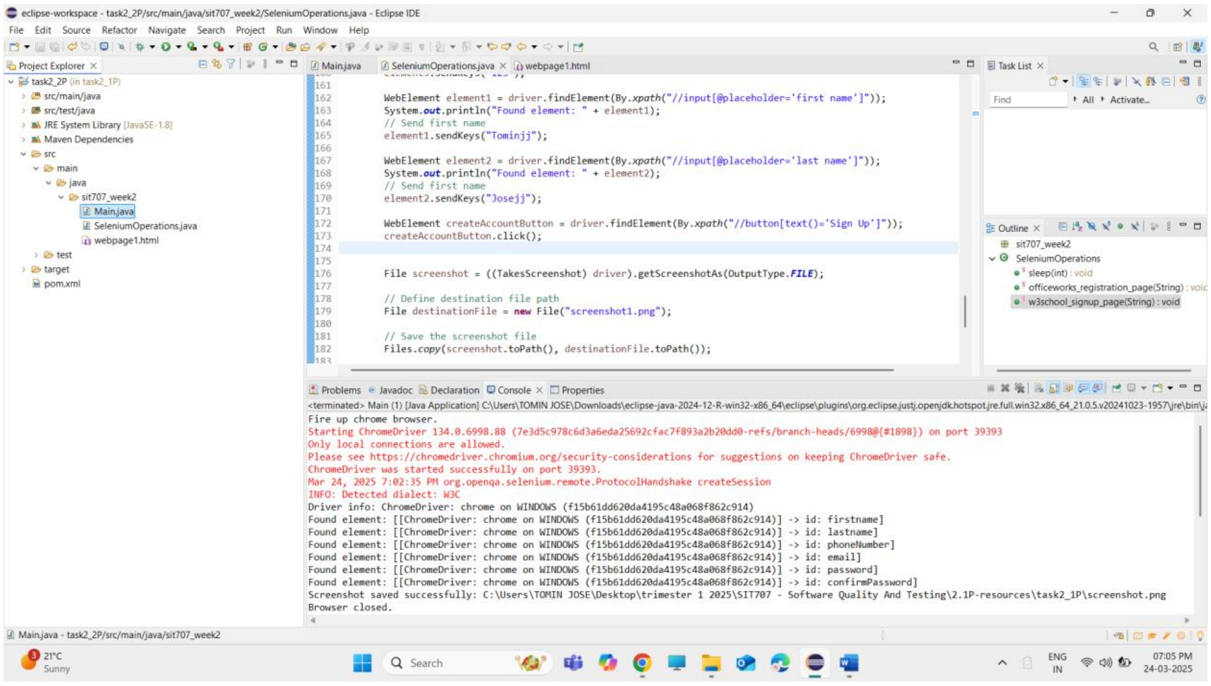
Confirm password: •••

What type of customer are you?

Personal | Business

By creating an account you're confirming you've read, understood and agree to the Officeworks **Privacy Policy** and **Terms of Use.**

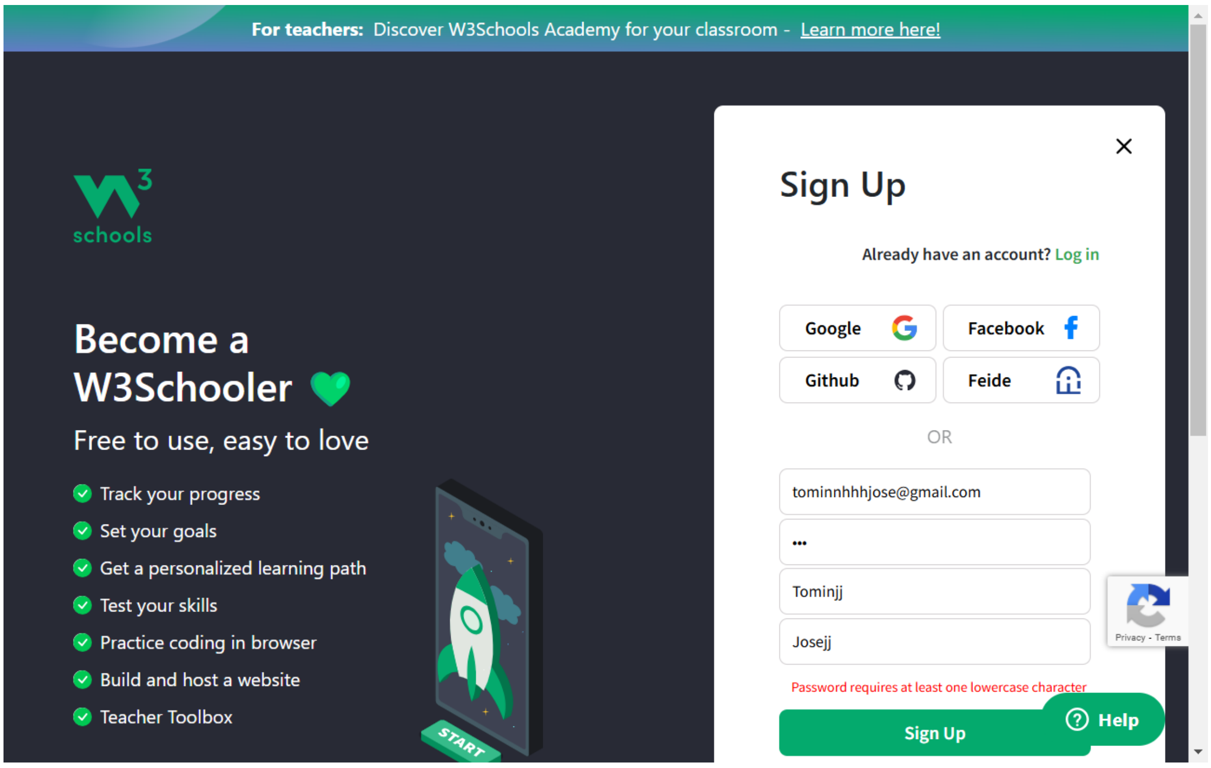
Output in the terminal:



```
        WebElement element1 = driver.findElement(By.xpath("//input[@placeholder='first name']"));
        System.out.println("Found element: " + element1);
        // Send first name
        element1.sendKeys("Tominjj");

        WebElement element2 = driver.findElement(By.xpath("//input[@placeholder='last name']"));
        System.out.println("Found element: " + element2);
        // Send first name
        element2.sendKeys("Josejj");

        WebElement createAccountButton = driver.findElement(By.xpath("//button[text()='Sign Up']"));
        createAccountButton.click();


        File screenshot = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);

        // Define destination file path
        File destinationFile = new File("screenshot1.png");

        // Save the screenshot file
        Files.copy(screenshot.toPath(), destinationFile.toPath());
```

```
<terminated> Main (1) [Java Application] C:\Users\TOMIN JOSE\Downloads\eclipse-java-2024-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\j
Fire up chrome browser.
Starting ChromeDriver 134.0.6998.88 (7e3d5c978c6d3a6eda25692cfac7f893a2b20dd0-refs/branch-heads/6998@{#1898}) on port 39393
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully on port 39393.
Mar 24, 2025 7:02:35 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Driver info: ChromeDriver: chrome on WINDOWS (f15b61dd620da4195c48a068f862c914)
Found element: [[ChromeDriver: chrome on WINDOWS (f15b61dd620da4195c48a068f862c914)] -> id: firstname]
Found element: [[ChromeDriver: chrome on WINDOWS (f15b61dd620da4195c48a068f862c914)] -> id: lastname]
Found element: [[ChromeDriver: chrome on WINDOWS (f15b61dd620da4195c48a068f862c914)] -> id: phoneNumber]
Found element: [[ChromeDriver: chrome on WINDOWS (f15b61dd620da4195c48a068f862c914)] -> id: email]
Found element: [[ChromeDriver: chrome on WINDOWS (f15b61dd620da4195c48a068f862c914)] -> id: password]
Found element: [[ChromeDriver: chrome on WINDOWS (f15b61dd620da4195c48a068f862c914)] -> id: confirmPassword]
Screenshot saved successfully: C:\Users\TOMIN JOSE\Desktop\trimester 1 2025\SIT707 - Software Quality And Testing\2.1P-resources\task2_1P\screenshot.png
Browser closed.
```

- **Repeat the above step for an alternative website's registration page.**
  I am using W3schools website for this.
  Screenshot:



For teachers: Discover W3Schools Academy for your classroom - Learn more here!

Become a W3Schooler

Free to use, easy to love

- Track your progress
- Set your goals
- Get a personalized learning path
- Test your skills
- Practice coding in browser
- Build and host a website
- Teacher Toolbox

Sign Up

Already have an account? Log in

Google | Facebook | Github | Feide

OR

tominnhhhjose@gmail.com

•••

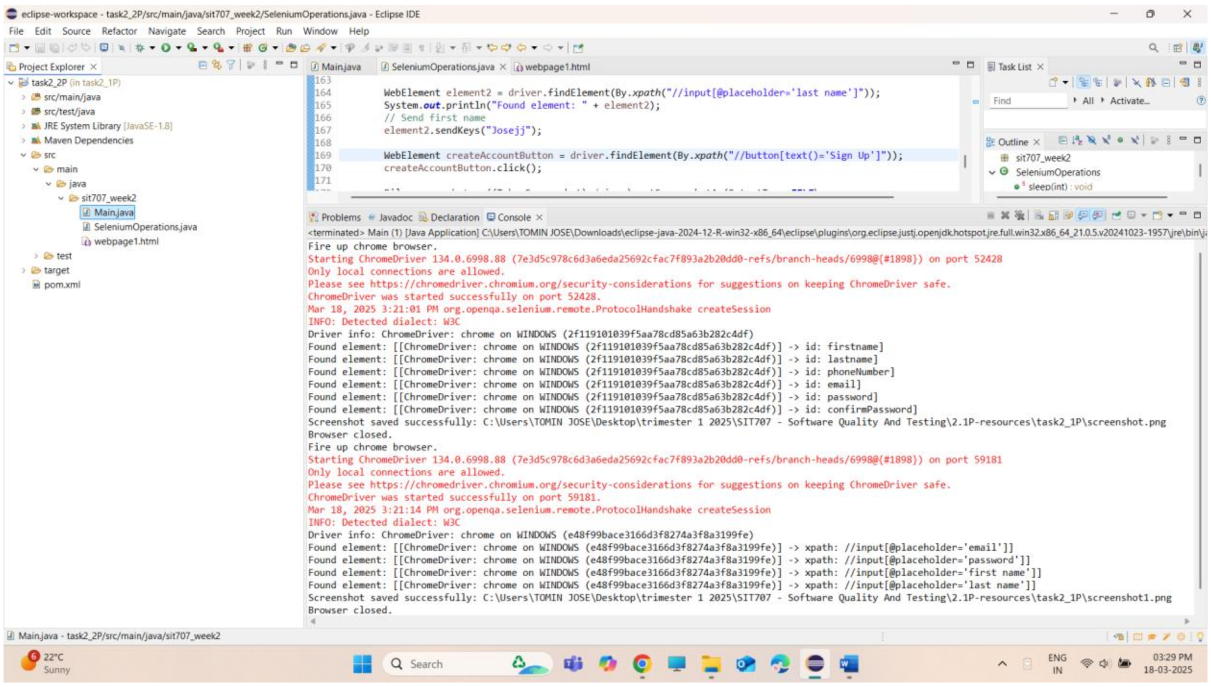
Tominjj

Josejj

Password requires at least one lowercase character

Sign Up

Output in terminal:



```
WebElement element2 = driver.findElement(By.xpath("//input[@placeholder='last name']"));
System.out.println("Found element: " + element2);
// Send first name
element2.sendKeys("Josejj");

WebElement createAccountButton = driver.findElement(By.xpath("//button[text()='Sign Up']"));
createAccountButton.click();
```

```
<terminated> Main (1) [Java Application] C:\Users\TOMIN JOSE\Downloads\eclipse-java-2024-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\j
Fire up chrome browser.
Starting ChromeDriver 134.0.6998.88 (7e3d5c978c6d3a6eda25692cfac7f893a2b20dd0-refs/branch-heads/6998@{#1898}) on port 52428
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully on port 52428.
Mar 18, 2025 3:21:01 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Driver info: ChromeDriver: chrome on WINDOWS (2f119101039f5aa78cd85a63b282c4df)
Found element: [[ChromeDriver: chrome on WINDOWS (2f119101039f5aa78cd85a63b282c4df)] -> id: firstname]
Found element: [[ChromeDriver: chrome on WINDOWS (2f119101039f5aa78cd85a63b282c4df)] -> id: lastname]
Found element: [[ChromeDriver: chrome on WINDOWS (2f119101039f5aa78cd85a63b282c4df)] -> id: phoneNumber]
Found element: [[ChromeDriver: chrome on WINDOWS (2f119101039f5aa78cd85a63b282c4df)] -> id: email]
Found element: [[ChromeDriver: chrome on WINDOWS (2f119101039f5aa78cd85a63b282c4df)] -> id: password]
Found element: [[ChromeDriver: chrome on WINDOWS (2f119101039f5aa78cd85a63b282c4df)] -> id: confirmPassword]
Screenshot saved successfully: C:\Users\TOMIN JOSE\Desktop\trimester 1 2025\SIT707 - Software Quality And Testing\2.1P-resources\task2_1P\screenshot.png
Browser closed.
Fire up chrome browser.
Starting ChromeDriver 134.0.6998.88 (7e3d5c978c6d3a6eda25692cfac7f893a2b20dd0-refs/branch-heads/6998@{#1898}) on port 59181
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully on port 59181.
Mar 18, 2025 3:21:14 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Driver info: ChromeDriver: chrome on WINDOWS (e48f99bace3166d3f8274a3f8a3199fe)
Found element: [[ChromeDriver: chrome on WINDOWS (e48f99bace3166d3f8274a3f8a3199fe)] -> xpath: //input[@placeholder='email']]
Found element: [[ChromeDriver: chrome on WINDOWS (e48f99bace3166d3f8274a3f8a3199fe)] -> xpath: //input[@placeholder='password']]
Found element: [[ChromeDriver: chrome on WINDOWS (e48f99bace3166d3f8274a3f8a3199fe)] -> xpath: //input[@placeholder='first name']]
Found element: [[ChromeDriver: chrome on WINDOWS (e48f99bace3166d3f8274a3f8a3199fe)] -> xpath: //input[@placeholder='last name']]
Screenshot saved successfully: C:\Users\TOMIN JOSE\Desktop\trimester 1 2025\SIT707 - Software Quality And Testing\2.1P-resources\task2_1P\screenshot1.png
Browser closed.
```

- **Highlight your key finding and experience dealing with registration page of 2 separate sites.**

While automating the Officeworks registration page, the experience was smooth due to the presence of consistent and stable id attributes for all form fields like firstname, lastname, email, etc. Locating elements using By.id() was simple and reliable. The "Create Account" button was easily identified through its data-testid attribute using a CSS selector. Overall, the page structure was clean and Selenium interactions were straightforward. After successfully submitting the form, a screenshot was captured and saved as screenshot.png, confirming the registration flow worked as expected.

On the other hand, automating the W3Schools signup page presented different challenges. The input fields lacked id attributes and relied on placeholder text, which required the use of XPath selectors for locating elements. The button had complex class names, so XPath based on button text (Sign Up) was a more stable approach.

- **Your program's source code (SeleniumOperations.java and the function you created for the alternative website).**
  SeleniumOperations.java:
  package sit707_week2;

  import org.openqa.selenium.By;
  import org.openqa.selenium.WebDriver;
  import org.openqa.selenium.WebElement;

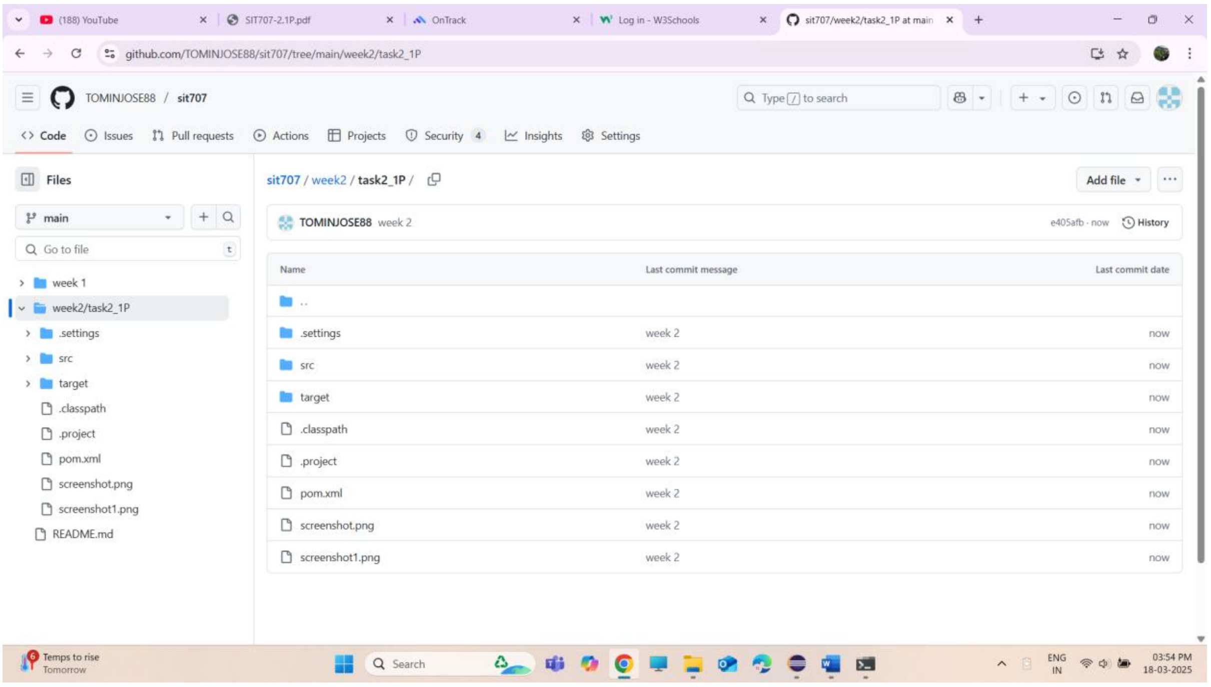
```
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.OutputType;

import java.io.File;
import java.io.IOException;
import java.nio.file.Files;


/**
 * This class demonstrates Selenium locator APIs to identify HTML elements.
 *
 * Details in Selenium documentation
https://www.selenium.dev/documentation/webdriver/elements/locators/
 *
 * @author Ahsan Habib
 */
public class SeleniumOperations {

	public static void sleep(int sec) {
		try {
			Thread.sleep(sec*1000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
	}


	public static void officeworks_registration_page(String url) {


		// Step 1: Locate chrome driver folder in the local drive.
		System.setProperty("webdriver.chrome.driver", "C:\\Users\\TOMIN
JOSE\\chromedriver-win64\\chromedriver.exe");

		// Step 2: Use above chrome driver to open up a chromium
browser.
		System.out.println("Fire up chrome browser.");
		WebDriver driver = new ChromeDriver();
```

```
            System.out.println("Driver info: " + driver);

            sleep(2);

            try {
            // Load a webpage in chromium browser.
            driver.get(url);

            /*
             * How to identify a HTML input field -
             * Step 1: Inspect the webpage,
             * Step 2: locate the input field,
             * Step 3: Find out how to identify it, by id/name/...
             */

            // Find first input field which is firstname
            WebElement element1 = driver.findElement(By.id("firstname"));
            System.out.println("Found element: " + element1);
            // Send first name
            element1.sendKeys("Tominjj");

            /*
             * Find following input fields and populate with values
             */
            // Write code
            WebElement element2 = driver.findElement(By.id("lastname"));
            System.out.println("Found element: " + element2);
            // Send first name
            element2.sendKeys("Josejj");

            WebElement element3 =
    driver.findElement(By.id("phoneNumber"));
            System.out.println("Found element: " + element3);
            // Send first name
            element3.sendKeys("0456456456");

            WebElement element4 = driver.findElement(By.id("email"));
            System.out.println("Found element: " + element4);
            // Send first name
            element4.sendKeys("tominnhhhjose@gmail.com");
            WebElement element5 = driver.findElement(By.id("password"));
            System.out.println("Found element: " + element5);
            // Send first name
            element5.sendKeys("123");

            WebElement element6 =
driver.findElement(By.id("confirmPassword"));
            System.out.println("Found element: " + element6);
            // Send first name
            element6.sendKeys("123");




            /*
             * Identify button 'Create account' and click to submit using
Selenium API.
             */
            WebElement createAccountButton =
driver.findElement(By.cssSelector("button[data-testid='account-action-btn']"));
            createAccountButton.click();


            sleep(5);

            /*
             * Take screenshot using selenium API.
             */
            // Capture screenshot as OutputType.FILE
    File screenshot = ((TakesScreenshot)
driver).getScreenshotAs(OutputType.FILE);

    // Define destination file path
    File destinationFile = new File("screenshot.png");

    // Save the screenshot file
    Files.copy(screenshot.toPath(), destinationFile.toPath());

    System.out.println("Screenshot saved successfully: " +
destinationFile.getAbsolutePath());
```

```
                    // Sleep a while
                    sleep(5);

                    // close chrome driver
                    driver.close();

                    } catch (IOException e) {
        System.out.println("There was an error saving the screenshot: " +
e.getMessage());
        e.printStackTrace();
     } finally {
          driver.quit();
        System.out.println("Browser closed.");
     }
          }

          public static void w3school_signup_page(String url) {

                    System.setProperty("webdriver.chrome.driver", "C:\\Users\\TOMIN
JOSE\\chromedriver-win64\\chromedriver.exe");

                    // Step 2: Use above chrome driver to open up a chromium
browser.
                    System.out.println("Fire up chrome browser.");
                    WebDriver driver = new ChromeDriver();

                    System.out.println("Driver info: " + driver);

                    sleep(2);

                    try {
                    // Load a webpage in chromium browser.
                    driver.get(url);

                    WebElement element4 =
driver.findElement(By.xpath("//input[@placeholder='email']"));
                    System.out.println("Found element: " + element4);
                    // Send first name
                    element4.sendKeys("tominnhhhjose@gmail.com");
                    WebElement element5 =
driver.findElement(By.xpath("//input[@placeholder='password']"));
                    System.out.println("Found element: " + element5);
                    // Send first name
                    element5.sendKeys("123");

                    WebElement element1 =
driver.findElement(By.xpath("//input[@placeholder='first name']"));
                    System.out.println("Found element: " + element1);
                    // Send first name
                    element1.sendKeys("Tominjj");

                    WebElement element2 =
driver.findElement(By.xpath("//input[@placeholder='last name']"));
                    System.out.println("Found element: " + element2);
                    // Send first name
                    element2.sendKeys("Josejj");

                    WebElement createAccountButton =
driver.findElement(By.xpath("//button[text()='Sign Up']"));
                    createAccountButton.click();


                    File screenshot = ((TakesScreenshot)
driver).getScreenshotAs(OutputType.FILE);

    // Define destination file path
    File destinationFile = new File("screenshot1.png");

    // Save the screenshot file
    Files.copy(screenshot.toPath(), destinationFile.toPath());

    System.out.println("Screenshot saved successfully: " +
destinationFile.getAbsolutePath());

    sleep(2);

                    }catch (IOException e) {
      System.out.println("There was an error saving the screenshot: " +
e.getMessage());
      e.printStackTrace();
```

```
        } finally {
            driver.quit();
          System.out.println("Browser closed.");
        }
            }


}
```

- **A screenshot of your GitHub page where your latest project folder is pushed.**



github.com/TOMINJOSE88/sit707/tree/main/week2/task2_1P

TOMINJOSE88 / sit707

sit707 / week2 / task2_1P /

TOMINJOSE88 week 2 e405afb · now History

Name	Last commit message	Last commit date
..		
.settings	week 2	now
src	week 2	now
target	week 2	now
.classpath	week 2	now
.project	week 2	now
pom.xml	week 2	now
screenshot.png	week 2	now
screenshot1.png	week 2	now

- **You should discuss the following key ideas in relation to your program while preparing the submission items:**
  • Selenium web-driver's locator APIs to identify input element and buttons (https://www.selenium.dev/documentation/webdriver/elements/locators)

  In the code, I utilized Selenium WebDriver's locator APIs to identify HTML elements such as input fields and buttons on two different web pages: Officeworks' registration page and W3Schools' signup page. For the Officeworks page, I used By.id to locate elements like firstname, lastname, phoneNumber, email, password and confirmPassword. This method is effective when elements have unique IDs, making it straightforward to interact with them. For the W3Schools page, I used By.xpath to locate elements by their placeholder attributes, such as email, password, first name, and last name. XPath is particularly useful when elements lack unique IDs or when more complex locators are needed. Additionally, I used By.cssSelector and By.xpath to locate and click the "Create account" and

"Sign Up" buttons, respectively. These locator APIs are essential for accurately identifying elements on a webpage, ensuring that the automation script interacts with the correct elements.

- **Selenium web-driver's element interaction APIs to interact with input element and buttons (https://www.selenium.dev/documentation/webdriver/elements/interactions).**

To interact with the identified elements, I employed Selenium WebDriver's element interaction APIs. For input fields, I used the sendKeys method to populate them with relevant data, such as names, phone numbers, emails, and passwords. This method simulates user input, allowing the script to fill out forms programmatically. For buttons, I used the click method to simulate user clicks, triggering form submissions or other actions. Additionally, I implemented the TakesScreenshot API to capture screenshots of the web pages after interacting with the elements. This is useful for debugging and verifying that the script is functioning as expected. Finally, I used sleep to introduce delays, ensuring that the webpage has enough time to load elements before interacting with them. These interaction APIs are crucial for automating user actions and validating the behavior of web applications.