**AUTOMATION TESTING OF ORANGEHRM WEBSITE USING SELENIUM WEBDRIVER**

**Group Members:**

**Huzaifa Kashif - (22K-5158)**

**Omer Ayaz Khan - (22K-5173)**

**Syed Yousif Ali - (22K-5174)**

**Selenium Webdriver:**

It is a powerful tool for automating web applications across various browsers, and provides a programming interface to interact with web elements on a web page.

**Types Of Testing Supported By Selenium:**

1. **Functional Testing:**

* Regression Testing
* User Acceptance Testing

1. **Non-Functional Testing:**

* Performance Testing
* Load Testing
* Stress Testing

1. **Compatibility Testing:**

* Cross-Browser Testing
* Cross-Platform Testing
* Device Testing

1. **Security Testing:**

* Vulnerability Testing
* Penetration Testing

1. **Usability Testing:**

* User Experience Testing

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| It supports major browsers, like Chrome, Safari, Firefox, Edge, allowing test cases to be executed across different environments. | It cannot be directly used for desktop, or mobile applications without integrating with other tools (e.g., Appium for mobile). |
| It supports multiple programming languages (like Java, C#, Python, Ruby, JavaScript, PHP) for writing test scripts. | Relies on external frameworks like TestNG for comprehensive reporting. |
| It interacts directly with the browser, mimicking real user interactions by performing actions at the OS level, providing robustness and increased performance. | Dealing with dynamic web elements (eg; that load asynchronously) requires additional handlings, like wait. |
| It can integrate with testing various frameworks and CI/CD tools like TestNG, Jenkins. | Unlike record-and-playback tools, Selenium WebDriver requires knowledge of a programming language, increasing the learning curve for non-programmers. |
| It is platform independent, meaning, it supports multiple operating systems, such as Windows, Linux, MacOS. | Selenium WebDriver is only suitable for web applications and cannot be used for desktop or mobile applications directly without integrating with other tools (e.g., Appium for mobile). |

**Application Selected For Automation Testing:**

**OrangeHRM (Demo Version) -** A web-based HR management system, commonly used for managing HR processes like user management, employee management, recruitment, attendance, and performance.

**Selection Criteria For Selenium Webdriver:**

1. **Application Type:**

OrangeHRM is a web-based HR management application, which makes it ideal to utilize Selenium Webdriver for functional testing.

1. **Open-Source:**

No licensing costs are required for using Selenium Webdriver.

1. **Community Support:**

Selenium Webdriver has a large community, providing ample tutorials, document and troubleshooting assistance.

1. **Requirement For Automation:**

For repetitive, and exhaustive functional and regression testing, automation with Selenium Webdriver helps save time, and reduce manual effort.

**Test Cases:**

|  |  |
| --- | --- |
| **Field** | **Description** |
| Name | autc-1: Verify that entering non-existent employee in Employee Name text-field would not add User to the system. |
| Requirement | R-1 (Add User) |
| Preconditions | 1. The user is at the user management admin page, and has pressed the ‘Add’ button. |
| Steps | 1. Provide valid entries for Username, Password, and Confirm Password text-fields. 2. Select either ‘Admin’, or ESS (Employee Self-Service) role from the user role drop-down menu. 3. Select either ‘Enabled’, or ‘Disabled’ status from the status drop-down menu. 4. Enter an employee that is not stored in the system for Employee Name text-field. 5. Click the Save button. |
| Expected Output: | 1. Red ‘Invalid’ word would become visible below the Employee Name text-field. 2. The user would not be added to the system. |

|  |  |
| --- | --- |
| **Field** | **Description** |
| Name | autc-2: Verify that leaving the Username text-field would not add User to the system. |
| Requirement | R-1 (Add User) |
| Preconditions | 1. The user is at the user management admin page, and has pressed the ‘Add’ button |
| Steps | 1. Provide valid entries for Password, and Confirm Password text-fields. 2. Select either ‘Admin’, or ESS (Employee Self-Service) role from the user role drop-down menu. 3. Select either ‘Enabled’, or ‘Disabled’ status from the status drop-down menu. 4. Enter an employee that is stored in the system for Employee Name text-field. 5. Click Save button |
| Expected Output: | 1. Red ‘Required’ word would become visible below the Username text-field. 2. The user would not be added to the system. |

|  |  |
| --- | --- |
| **Field** | **Description** |
| Name | autc-4: Verify that entering all valid data for all text-fields in the Add User page would add a new OrangeHRM system user. |
| Requirement | R-1 (Add User) |
| Preconditions | 1. The user is at the user management admin page, and has pressed the ‘Add’ button |
| Steps | 1. Provide valid entries for Username, Password, and Confirm Password text-fields. 2. Select either ‘Admin’, or ESS (Employee Self-Service) role from the user role drop-down menu. 3. Select either ‘Enabled’, or ‘Disabled’ status from the status drop-down menu. 4. Enter an employee that is stored in the system for Employee Name text-field. 5. Click Save button |
| Expected Output: | 1. A green prompt will appear at the bottom left of the screen, stating, “User Added” 2. The user would be added to the system. |

|  |  |
| --- | --- |
| **Field** | **Description** |
| Name | sutc-2: Verify that entering all valid data for all text-fields in the Search System Users part of admin User Management page would retrieve the required user’s data |
| Requirement | R-2 (Search User) |
| Preconditions | 1. The user is at the user management admin page. |
| Steps | 1. Provide valid entry for Username text-field. 2. Select either ‘Admin’, or ESS (Employee Self-Service) role from the user role drop-down menu. 3. Select either ‘Enabled’, or ‘Disabled’ status from the status drop-down menu. 4. Enter an employee that is stored in the system for Employee Name text-field. 5. Click the Search button. |
| Expected Output: | 1. The required user’s data would be retrieved.. |

**Test Scripts:**

**Pomautomate.java:**

package TestProject;  
  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.support.ui.ExpectedConditions;  
import org.openqa.selenium.support.ui.Select;  
import org.openqa.selenium.support.ui.WebDriverWait;  
import org.testng.Assert;  
  
import java.time.Duration;  
import java.util.List;  
import java.util.concurrent.TimeUnit;  
  
public class POMautomateClass {  
  
 public void pageTitle(WebDriver obj){  
 //to print the page title on the console.  
 System.*out*.println("\nPage Title: ");  
 String pagetitle= obj.getTitle();  
 System.*out*.println(pagetitle);  
 }  
  
 public void loginPage(WebDriver obj){ //instead of passing obj of class Webdriver as a  
 //parameter for every function within this class, I can also pass obj of class Webdriver  
 //only once in this class's parameterized constructor.  
  
 System.*out*.println("\nLogin Page URL:");  
 String pageurl= obj.getCurrentUrl();  
 System.*out*.println(pageurl);  
  
 //Login  
 obj.findElement(By.*name*("username")).sendKeys("Admin");  
 obj.findElement(By.*name*("password")).sendKeys("admin123");  
 obj.findElement(By.*xpath*("//button[@type='submit']")).click();  
 //using assertion to confirm the login success via dashboard text.  
  
 String loginsuccessviatext=obj.findElement(By.*xpath*("//div[@class='oxd-topbar-header-title']")).getText();  
  
 if (loginsuccessviatext.equals("Dashboard")){  
 System.*out*.println("Login Successful.\nCommencing Operations...");  
 }  
 else {  
 System.*out*.println("Login Unsuccessful");  
 }  
  
 // Aside from using the login successful if-else condition for verification (lines 32-37) in the console,  
 // we can also use TestNG Assertions which are more efficient.  
  
 Assert.*assertEquals*(loginsuccessviatext,"Dashboard");  
 }  
  
 public void addUserTest(WebDriver obj, String userRole, String employeeNameExisting, String userName,String status, String password, String confirmPassword, String button){  
 try{  
 List<WebElement> dropDown = obj.findElements(By.*xpath*("//i[@class='oxd-icon bi-caret-down-fill oxd-select-text--arrow']"));  
  
 //for userRole;  
 // Click on the dropdown arrow to open the dropdown menu  
  
 if (userRole.equals("ESS")){  
 // Click on the dropdown arrow to open the dropdown menu  
  
 // Open the User Role dropdown  
 dropDown.get(0).click();  
 Thread.*sleep*(2000); // Wait briefly to ensure the dropdown is fully expanded  
  
 // Select the 'ESS' option from the dropdown  
 WebElement essOption = obj.findElement(By.*xpath*("//div[@role='option']//span[text()='ESS']"));  
 essOption.click();  
 }else if (userRole.equals("Admin")){  
  
 // Open the User Role dropdown  
 dropDown.get(0).click();  
 Thread.*sleep*(2000); // Wait briefly to ensure the dropdown is fully expanded  
  
 WebElement adminOption = obj.findElement(By.*xpath*("//div[@role='option']//span[text()='Admin']"));  
 adminOption.click();  
 }  
  
 //for Employee Name  
 obj.findElement(By.*xpath*("//input[@placeholder='Type for hints...']")).clear();  
 obj.findElement(By.*xpath*("//input[@placeholder='Type for hints...']")).sendKeys(employeeNameExisting);  
  
  
 Thread.*sleep*(6000);  
  
 WebElement firstOption = obj.findElement(By.*xpath*("//div[contains(@class, 'oxd-autocomplete-option')]"));  
 firstOption.click();  
  
 //for Username, Password, and Confirm Password textfields, which have no unique attribute among them, and are identified by the same xpath, I will use List<>;  
  
 // Locate all three input fields using the shared XPath  
 List<WebElement> inputFields = obj.findElements(By.*xpath*("//div[@class='oxd-input-group oxd-input-field-bottom-space']//input[@class='oxd-input oxd-input--active']"));  
  
 // Type into each field by accessing them through their index  
 // Username field (first input element)  
 inputFields.get(0).sendKeys(userName);  
  
 // Password field (second input element)  
 inputFields.get(1).sendKeys(password);  
  
 // Confirm Password field (third input element)  
 inputFields.get(2).sendKeys(confirmPassword);  
  
  
 // for userName;  
 // obj.findElement(By.className("oxd-input oxd-input--active")).clear();  
 // obj.findElement(By.className("oxd-input oxd-input--active")).sendKeys(userName);  
  
 //for password  
 // obj.findElement(By.className("oxd-input oxd-input--active")).clear();  
 // obj.findElement(By.className("oxd-input oxd-input--active")).sendKeys(password);  
  
 //for confirm password  
 // obj.findElement(By.className("oxd-input oxd-input--active")).clear();  
 // obj.findElement(By.className("oxd-input oxd-input--active")).sendKeys(confirmPassword);  
  
 //for status  
 if (status.equals("Enabled")){  
 // Click on the dropdown arrow to open the dropdown menu  
  
 // Open the status dropdown  
 // obj.findElement(By.xpath("//i[@class='oxd-icon bi-caret-down-fill oxd-select-text--arrow']")).click();  
  
 dropDown.get(1).click();  
 Thread.*sleep*(2000); // Wait briefly to ensure the dropdown is fully expanded  
  
 // Select the 'Enabled' option from the dropdown  
 WebElement enabledOption = obj.findElement(By.*xpath*("//div[@role='option']//span[text()='Enabled']"));  
 enabledOption.click();  
  
 }else if (status.equals("Disabled")){  
  
 // Open the User Role dropdown  
 // obj.findElement(By.xpath("//i[@class='oxd-icon bi-caret-down-fill oxd-select-text--arrow']")).click();  
 dropDown.get(1).click();  
 Thread.*sleep*(2000); // Wait briefly to ensure the dropdown is fully expanded  
  
 WebElement disabledOption = obj.findElement(By.*xpath*("//div[@role='option']//span[text()='Disabled']"));  
 disabledOption.click();  
 }  
  
 if (button.equals("Save")){  
  
 Thread.*sleep*(3000);  
 //for save button  
 obj.findElement(By.*xpath*("//button[@type='submit']")).click();  
 }  
 else if(button.equals("Cancel")){  
 Thread.*sleep*(3000);  
 // for cancel button  
 obj.findElement(By.*xpath*("//button[@class='oxd-button oxd-button--medium oxd-button--ghost']")).click();  
 }  
  
 obj.manage().timeouts().implicitlyWait(3, TimeUnit.*SECONDS*);  
 List<WebElement> errorMessages = obj.findElements(By.*xpath*("//span[contains(@class, 'oxd-text oxd-text--span oxd-input-field-error-message oxd-input-group\_\_message')]"));  
  
 if (errorMessages.isEmpty()){  
 System.*out*.println("Add User: "+userName+" -- Successful");  
 }else{  
 System.*out*.println("Add User: "+userName+" -- Unsuccessful");  
  
 for (WebElement errorMessage:errorMessages){  
 System.*out*.println(errorMessage.getText());  
 }  
 }  
  
 }catch(Exception e){  
 System.*out*.println("An unexpected error occurred while adding User: "+userName);  
 }  
 }  
  
 public void searchUserTest(WebDriver obj, String userRole, String employeeNameExisting, String userName,String status, String button){  
 try{  
  
 //for User Role  
 List<WebElement> dropDown = obj.findElements(By.*xpath*("//i[@class='oxd-icon bi-caret-down-fill oxd-select-text--arrow']"));  
  
 if (userRole.equals("ESS")){  
  
 // Open the User Role dropdown  
 dropDown.get(0).click();  
 Thread.*sleep*(2000); // Wait briefly to ensure the dropdown is fully expanded  
  
 // Select the 'ESS' option from the dropdown  
 WebElement essOption = obj.findElement(By.*xpath*("//div[@role='option']//span[text()='ESS']"));  
 essOption.click();  
 }else if (userRole.equals("Admin")){  
  
 // Open the User Role dropdown  
 dropDown.get(0).click();  
 Thread.*sleep*(2000); // Wait briefly to ensure the dropdown is fully expanded  
  
 WebElement adminOption = obj.findElement(By.*xpath*("//div[@role='option']//span[text()='Admin']"));  
 adminOption.click();  
 }  
  
 //Username  
 //Search and Username in the search area share the same xpath, and no unique attributes. Hence, using List<>  
 List<WebElement> inputFields = obj.findElements(By.*xpath*("//input[@class='oxd-input oxd-input--active']"));  
  
 // Type into userName field by accessing it through it index  
 // Username field (second input index element. First input index element is search)  
 inputFields.get(1).sendKeys(userName);  
  
 //for status  
 if (status.equals("Enabled")){  
 // Click on the dropdown arrow to open the dropdown menu  
  
 // Open the status dropdown  
 dropDown.get(1).click();  
 Thread.*sleep*(2000); // Wait briefly to ensure the dropdown is fully expanded  
  
 // Select the 'Enabled' option from the dropdown  
 WebElement enabledOption = obj.findElement(By.*xpath*("//div[@role='option']//span[text()='Enabled']"));  
 enabledOption.click();  
  
 }else if (status.equals("Disabled")){  
  
 // Open the User Role dropdown  
 dropDown.get(1).click();  
 Thread.*sleep*(2000); // Wait briefly to ensure the dropdown is fully expanded  
  
 WebElement disabledOption = obj.findElement(By.*xpath*("//div[@role='option']//span[text()='Disabled']"));  
 disabledOption.click();  
 }  
  
 //for Employee Name  
 obj.findElement(By.*xpath*("//input[@placeholder='Type for hints...']")).clear();  
 obj.findElement(By.*xpath*("//input[@placeholder='Type for hints...']")).sendKeys(employeeNameExisting);  
  
 Thread.*sleep*(6000); // allows the program to wait for the drop-down to completely expand before selectinf the first-most option  
  
 WebElement firstOption = obj.findElement(By.*xpath*("//div[contains(@class, 'oxd-autocomplete-option')]"));  
 firstOption.click();  
  
 if (button.equals("Search")){  
 Thread.*sleep*(2000);  
 //for search button  
 obj.findElement(By.*xpath*("//button[@type='submit']")).click();  
 Thread.*sleep*(5000);  
 // reset search bar via reset button after searching for the required record  
 obj.findElement(By.*xpath*("//button[@class='oxd-button oxd-button--medium oxd-button--ghost']")).click();  
  
 }  
 else if(button.equals("Reset")){  
 Thread.*sleep*(2000);  
 // for reset button  
 obj.findElement(By.*xpath*("//button[@class='oxd-button oxd-button--medium oxd-button--ghost']")).click();  
 }  
  
 }catch (Exception e){  
 System.*out*.println("An unexpected error occurred while searching for an existing User: "+userName);  
 }  
 }  
  
 public void logoutPage(WebDriver obj) throws InterruptedException {  
 //Logout  
 obj.findElement(By.*xpath*("//span[@class='oxd-userdropdown-tab']")).click();  
 Thread.*sleep*(2000); //for slowing down the execution of the next line of code.  
 obj.findElement(By.*linkText*("Logout")).click();  
  
 String logoutURL=obj.getCurrentUrl();  
  
 if (logoutURL.contains("/login")){  
 System.*out*.println("Logout Successful.");  
 }else{  
 System.*out*.println("Logout Unsuccessful.");  
 }  
  
 // Aside from using the logout successful if-else condition for verification (lines 55-59) in the console,  
 // we can also use TestNG Assertions which are more efficient.  
  
 Assert.*assertTrue*(logoutURL.contains(("/login")));  
  
 System.*out*.println("\nAutomation Successful.");  
  
 }  
}

**TestNGTest.java:**

package TestProject.tests\_Testng;  
  
import TestProject.POMautomateClass;  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
import org.openqa.selenium.support.ui.WebDriverWait;  
import org.testng.annotations.AfterClass;  
import org.testng.annotations.BeforeClass;  
import org.testng.annotations.Test;  
import util.configUtil;  
  
import java.time.Duration;  
import java.util.Properties;  
import java.util.concurrent.TimeUnit;  
  
public class TestNGTest {  
 POMautomateClass pom;  
 WebDriver driver;  
  
 @BeforeClass //@BeforeClass: Sets up the WebDriver and opens the login page.  
 public void initiatePage(){  
 System.*setProperty*("webdriver.chrome.driver", "F:\\chromedriver-win64\\chromedriver.exe");  
  
 driver=new ChromeDriver();  
  
 // Page-Object Model Approach  
  
 pom=new POMautomateClass();  
  
 //I used this to wait for the username and the password when the login field  
 // is automated.  
  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*); // to solve synchronization issues with the website being automated.  
  
 driver.manage().window().maximize();  
 driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");  
  
 pom.pageTitle(driver);  
 pom.loginPage(driver);  
  
 }  
  
 //use priority to make it clear that which test-case should execute first,  
 // and which test-case should follow it.  
  
  
 @Test (description = "Add User1" ,priority = 1)  
 public void addUserTestCase1(){  
  
 driver.findElement(By.*className*("oxd-main-menu-item")).click();  
 driver.findElement(By.*xpath*("//i[@class='oxd-icon bi-plus oxd-button-icon']")).click();  
  
 System.*out*.println("\n\n//Invalid: Non-existent employee.");  
 // Invalid: Non-existent employee  
 pom.addUserTest(driver, "Admin", "Huzaifa Kashif", "newUser3", "Enabled", "Password123!", "Password123!", "Save");  
  
 //pom.addUserTest(driver,"Admin","Timothy Lewis Amiano", "Timothy1","Enabled","Charles@123","Charles@123", "Save");  
  
 //To navigate back to the Add User page to execute the next test case.  
  
 // driver.findElement(By.className("oxd-main-menu-item active")).click();  
 // driver.findElement(By.xpath("//i[@class='oxd-icon bi-plus oxd-button-icon']")).click();  
  
 /\* pom.addUserTest(driver, "Admin", "John Smith", "", "Enabled", "Password123!", "Password123!", "Save"); // Invalid: Empty username  
  
 driver.findElement(By.className("oxd-main-menu-item active")).click();  
 driver.findElement(By.xpath("//i[@class='oxd-icon bi-plus oxd-button-icon']")).click();  
  
 pom.addUserTest(driver, "Admin", "John Smith", "newUser2", "Enabled", "pass", "pass", "Save"); // Invalid: Weak password  
  
 driver.findElement(By.className("oxd-main-menu-item active")).click();  
 driver.findElement(By.xpath("//i[@class='oxd-icon bi-plus oxd-button-icon']")).click();  
  
 pom.addUserTest(driver, "Admin", "Invalid Employee", "newUser3", "Enabled", "Password123!", "Password123!", "Save"); // Invalid: Non-existent employee  
  
 \*/  
 }  
  
 @Test (description = "Add User2" ,priority = 2)  
 public void addUserTestCase2(){  
 driver.findElement(By.*className*("oxd-main-menu-item")).click();  
 driver.findElement(By.*xpath*("//i[@class='oxd-icon bi-plus oxd-button-icon']")).click();  
  
 System.*out*.println("//Invalid: Empty username.");  
 // Invalid: Empty username  
 pom.addUserTest(driver, "Admin", "Charles Carter", "", "Enabled", "Password123!", "Password123!", "Save");  
 }  
  
 @Test (description = "Add User3" ,priority = 3)  
 public void addUserTestCase3(){  
 driver.findElement(By.*className*("oxd-main-menu-item")).click();  
 driver.findElement(By.*xpath*("//i[@class='oxd-icon bi-plus oxd-button-icon']")).click();  
  
 System.*out*.println("//Invalid add user test case: Password and Confirm Password do not match.");  
 //Invalid add user test case: Password and Confirm Password do not match  
 pom.addUserTest(driver,"Admin","Charles Carter", "Timothy1","Enabled","Charles@123","Charles@12", "Save");  
 }  
  
 @Test (description = "Add User4" ,priority = 4)  
 public void addUserTestCase4(){  
 driver.findElement(By.*className*("oxd-main-menu-item")).click();  
 driver.findElement(By.*xpath*("//i[@class='oxd-icon bi-plus oxd-button-icon']")).click();  
  
 System.*out*.println("//Valid add user test case: Add a new User..");  
 //Valid add user test case: Add a new User.  
 pom.addUserTest(driver,"Admin","Charles Carter", "Timothy1","Enabled","Charles@123","Charles@123", "Save");  
 }  
  
 @Test (description = "Add User5" ,priority = 5)  
 public void addUserTestCase5(){  
 driver.findElement(By.*className*("oxd-main-menu-item")).click();  
 driver.findElement(By.*xpath*("//i[@class='oxd-icon bi-plus oxd-button-icon']")).click();  
  
 System.*out*.println("Invalid add user test case: User already exists.");  
 //Invalid add user test case: User already exists  
 pom.addUserTest(driver,"Admin","Charles Carter", "Timothy1","Enabled","Charles@123","Charles@123", "Save");  
 }  
  
 @Test (description = "Search User1" ,priority = 6)  
 public void searchUserTestCase1() {  
 driver.findElement(By.*className*("oxd-main-menu-item")).click();  
 // driver.findElement(By.xpath("//i[@class='oxd-icon bi-plus oxd-button-icon']")).click();  
  
 System.*out*.println("\n\n//valid search user test case: find existing user by passing all parameters.");  
 //valid search user test case: find existing user by passing all parameters  
 pom.searchUserTest(driver, "Admin","Charles Carter","","","Search");  
  
 }  
  
 @Test (description = "Search User2" ,priority = 7)  
 public void searchUserTestCase2(){  
 driver.findElement(By.*className*("oxd-main-menu-item")).click();  
 // driver.findElement(By.xpath("//i[@class='oxd-icon bi-plus oxd-button-icon']")).click();  
  
 System.*out*.println("//valid search user test case: find existing user by passing all parameters.");  
 //valid search user test case: find existing user by passing all parameters  
 pom.searchUserTest(driver, "Admin","Charles Carter","Timothy1","Enabled","Search");  
 }  
  
 @AfterClass //@AfterClass: Closes the WebDriver after a short delay.  
 public void closePage() throws InterruptedException {  
 Thread.*sleep*(5000);  
 //driver.manage().timeouts().implicitlyWait(8, TimeUnit.SECONDS);  
 pom.logoutPage(driver);  
 Thread.*sleep*(3000);  
 driver.close();  
 }  
}

**RunnerFile.xml:**

<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name="Smoke suite">

<test name="TrainingPractice">

<classes>

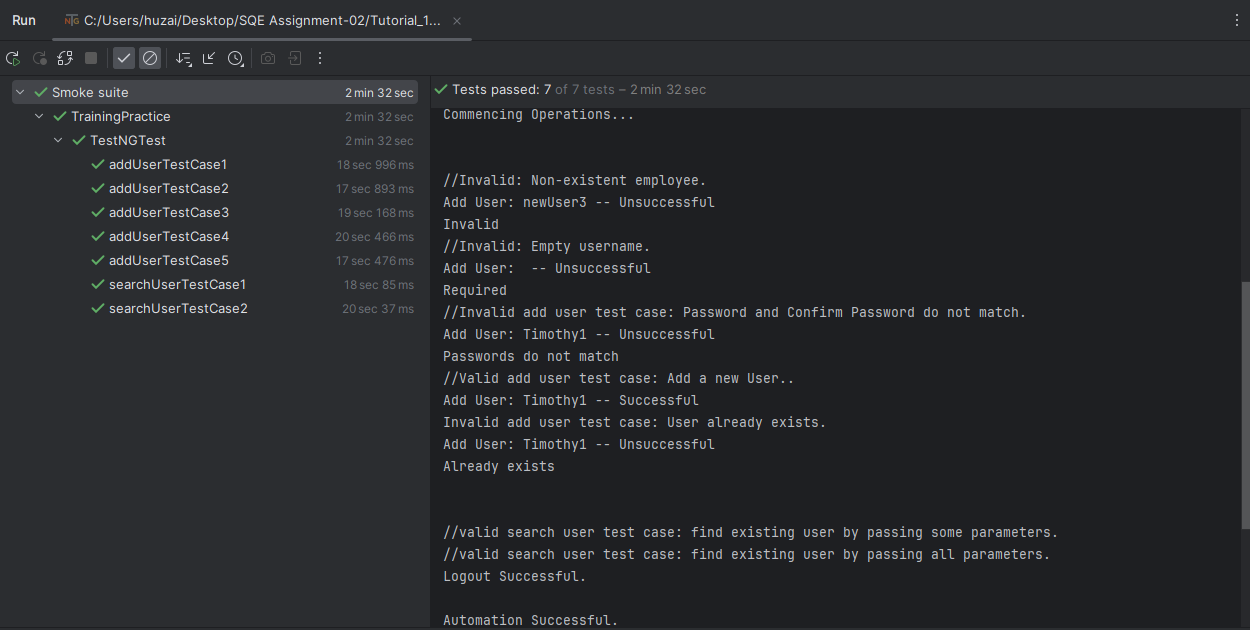
<class name="TestProject.tests\_Testng.TestNGTest"></class>

</classes>

</test>

</suite>

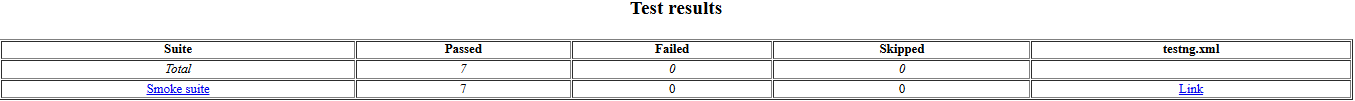
**Test Case Output:**

****

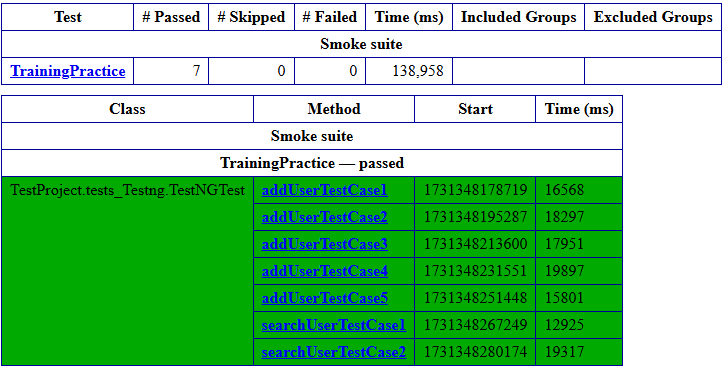
* **TrainingPractice.html:**

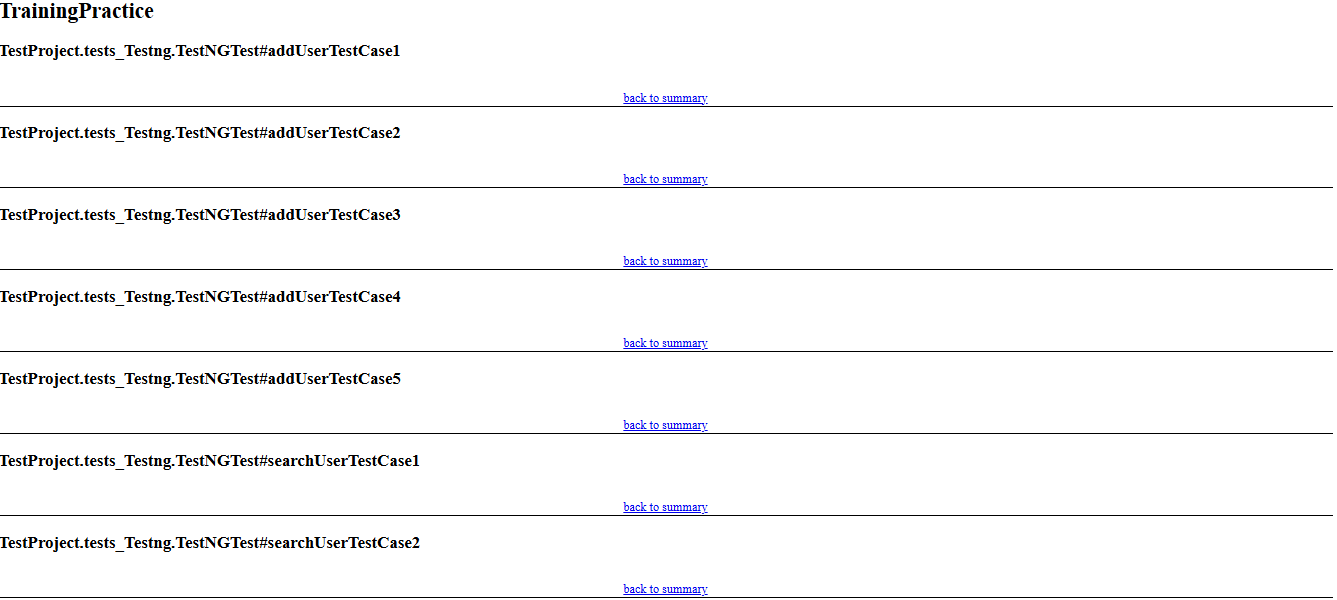
****

* **Index.html:**

****

* **emailable-report.html:**

****

****

Test Case Execution (Video) is included in PowerPoint Presentation. Cannot include it in Word.