**Overview**

Your code represents a **TestNG**-based automation framework using **Selenium WebDriver**. The code is structured to test the login functionality of the OrangeHRM application (https://opensource-demo.orangehrmlive.com/web/index.php/auth/login).

The framework consists of the following components:

1. **BrowserTest class**: Sets up and tears down the browser for each test case.
2. **LoginPage class**: Encapsulates the page elements and actions related to the login page using the Page Object Model (POM).
3. **LoginTest class**: Contains test cases to validate login functionality.

**1. BrowserTest Class**

This is a base class that:

1. Sets up the browser before each test (@BeforeMethod).
2. Tears down the browser after each test (@AfterMethod).

**Key Elements**

* **Browser Setup**

java

Copy code

@BeforeMethod

@Parameters("browser")

public void setUp(@Optional("chrome") String browser)

* + Uses the **@Parameters annotation** to pass the browser type from the TestNG XML configuration.
  + If no parameter is provided, Chrome is used by default (using @Optional).
  + Sets up drivers for Chrome, Firefox, or Edge using the **WebDriverManager** library to avoid manual driver management.
* **Maximize Browser and Navigate to URL**

java

Copy code

driver.manage().window().maximize();

driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

* + Maximizes the browser window.
  + Opens the login page of OrangeHRM.
* **Browser Cleanup**

java

Copy code

@AfterMethod

public void tearDown() {

if (driver != null) {

driver.quit();

}

}

* + Ensures the browser is closed after each test, releasing resources.

**2. LoginPage Class**

This class encapsulates the login page logic and uses the **Page Object Model (POM)** pattern.

**Key Elements**

* **Locators**

java

Copy code

private By emailField = By.xpath(...);

private By passwordField = By.xpath(...);

* + Locates elements using XPath expressions.
  + Each element (e.g., email field, password field, login button) is represented by a By locator.
* **Constructor**

java

Copy code

public LoginPage(WebDriver driver) {

this.driver = driver;

}

* + Accepts a WebDriver instance and associates it with this page.
* **Actions**
  + enterEmail(String username) and enterPassword(String password):

java

Copy code

driver.findElement(emailField).sendKeys(username);

* + - Sends text to the email and password fields.
  + clickLogin():

java

Copy code

driver.findElement(loginButton).click();

* + - Clicks the login button.
  + passwordreset():

java

Copy code

driver.findElement(passwordResetButton).click();

* + - Clicks the "Forgot Password" link.
* **Error Messages**

java

Copy code

public WebElement getErrorMessage() {

return driver.findElement(errorMessage);

}

Retrieves error messages displayed for validation failures.

**3. LoginTest Class**

This class contains test methods to validate the login functionality using **TestNG**.

**Key Elements**

* **Test Case: Valid Login**

java

Copy code

@Test(priority = 1)

public void testValidLogin() {

loginPage.enterEmail("Admin");

loginPage.enterPassword("admin123");

loginPage.clickLogin();

}

* + Enters valid credentials (Admin/admin123) and clicks login.
  + Uses **WebDriverWait** to wait until the dashboard URL contains /dashboard/index.
  + Asserts the current URL matches the expected dashboard URL:

java

Copy code

Assert.assertEquals(driver.getCurrentUrl(), expectedUrl);

* **Test Case: Empty Fields**

java

Copy code

@Test(priority = 3)

public void testEmptyEmailOrMobileFields() {

loginPage.clickLogin();

WebElement errorBox = loginPage.getErrorMessage();

}

* + Attempts login without entering credentials.
  + Asserts that the "Required" error message is displayed.
* **Test Case: Invalid Login**

java

Copy code

@Test(priority = 2)

public void testInvalidLogin() {

loginPage.enterEmail("invalidUser@example.com");

loginPage.clickLogin();

}

* + Enters an invalid email and asserts that the appropriate error message is displayed.
* **Test Case: Password Reset Flow**

java

Copy code

@Test

public void testMailResetFlow() {

loginPage.passwordreset();

}

* + Clicks the "Forgot Password" link and validates the password reset page.

**Tools and Concepts Used**

1. **TestNG Framework**
   * Test annotation (@Test), priority (priority), and parameterization (@Parameters).
   * Assertions (Assert.assertEquals, Assert.assertTrue).
2. **Selenium WebDriver**
   * Locates elements using **XPath**.
   * Performs actions like sendKeys, click, and getText.
3. **Page Object Model (POM)**
   * Separates the test logic (LoginTest) from page-specific details (LoginPage).
4. **WebDriverManager**
   * Simplifies browser driver setup.
5. **Error Handling**
   * Validates error messages for empty fields and invalid inputs.
6. **URL Validation**
   * Ensures the user is redirected to the correct page upon successful login.

**Execution Flow**

1. The BrowserTest class initializes the browser and navigates to the OrangeHRM login page.
2. The LoginTest class interacts with the LoginPage class to perform actions and validations for different scenarios:
   * Valid login.
   * Empty field validation.
   * Invalid login.
   * Password reset.
3. After each test, the browser is closed.