Create 2 test cases, disable one using enabled = false, and run only the active test.

Write a test to run the same test multiple times.

Write test cases for a dummy login page using @Parameters in testng.xml.

Write dependent test cases:

login()

search Product() (depends on login)

logout() (depends on search)

5.Use Data Provider to supply multiple sets of usernames/passwords to a login test.

6.Run test cases in parallel (methods, classes, tests) using parallel attribute in testng.xml.

Answers

Code:

```
//1.
package Testng;
import org.testng.annotations.Test;
public class enableddisable{
@Test(enabled = true)
public void activeTest() {
System.out.println("This is the active test running");
}
@Test(enabled = false)
public void disabledTest() {
System.out.println("This test won't run as it's disabled");
```

```
}
}
//2.
package Testng;
import org.testng.annotations.Test;
public class multipleruns {
@Test
public void repeatThisTest() {
int numberOfRuns = 5; // Set how many times you want to run the test logic
for (int i = 0; i < numberOfRuns; i++) {
System.out.println("This test is running multiple times - " + System.currentTimeMillis());
}
}
}

//3.
package Testng;
import org.testng.annotations.Optional;
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;
import static org.testng.Assert.*;
public class loginpage {
@Test
@Parameters({"username", "password"})
public void test(@Optional("admin") String user,@Optional("admin@123") String pass)
{
```

```
System.out.println("Testing login with:");
System.out.println("Username: " + user);
System.out.println("Password: " + pass);
// Actual login validation logic
if(user.equals("admin") && pass.equals("admin@123")) {
System.out.println("Successfully logged in");
assertTrue(true, "Valid credentials should login successfully");
} else {
System.out.println("Login failed");
assertFalse(user.equals("admin") && pass.equals("admin@123"),
"Invalid credentials should fail login");
}
}
}
```

```
//4.
package Testng;
import java.time.Duration;
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.interactions.Actions;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.annotations.AfterClass;
```

```
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;
public class dependent {
WebDriver driver;
private WebDriverWait wait;
String baseUrl = "https://www.ebay.com/";
@BeforeClass
public void setup() {
driver = new ChromeDriver();
driver.manage().window().maximize();
wait = new WebDriverWait(driver, Duration.ofSeconds(15));
}
@Test(priority = 1, description = "Login to eBay account")
public void login() {
driver.get(baseUrl);
WebElement signInLink = wait.until(ExpectedConditions.elementToBeClickable(
By.linkText("Sign in")));
signInLink.click();

WebElement usernameField = wait.until(ExpectedConditions.visibilityOfElementLocated(
By.id("userid")));
usernameField.sendKeys("username951520@gmail.com");
WebElement continueButton = wait.until(ExpectedConditions.elementToBeClickable(
By.id("signin-continue-btn")));
continueButton.click();
```

```
WebElement passwordField =
wait.until(ExpectedConditions.elementToBeClickable(By.id("pass")));

passwordField.sendKeys("Ammanana@0310");

WebElement signInButton = wait.until(ExpectedConditions.elementToBeClickable(

By.id("sgnBt")));

signInButton.click();


System.out.println("Login steps completed");

}


@Test(dependsOnMethods = "login")

public void searchProduct() {

WebElement searchBox = driver.findElement(By.id("gh-ac"));

searchBox.sendKeys("iPhone 13");

driver.findElement(By.id("gh-search-btn")).click();

System.out.println("Product search completed");

}

@Test(dependsOnMethods = "searchProduct")

public void logout() {

WebElement accountMenu = driver.findElement(By.cssSelector("#gh > nav > div.gh-nav__left-
wrap > span.gh-identity")); ;

Actions actions = new Actions(driver);

actions.moveToElement(accountMenu).build().perform();

WebElement signOutLink = wait.until(ExpectedConditions.elementToBeClickable(

By.linkText("Sign out")));

signOutLink.click();
```

```
System.out.println("Logout completed");
}
@AfterClass
public void Test() {
if (driver != null) {
driver.close();
}
}
}

//5.

package Testng;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;
public class dataprovider {
WebDriver driver;
@BeforeMethod
public void setUp() {
driver = new ChromeDriver();
```

```
driver.get("https://www.saucedemo.com/");
}
@DataProvider(name = "loginData")
public Object[][] provideLoginData() {
return new Object[][] {
{"standard_user", "secret_sauce", true}, // valid credentials
{"locked_out_user", "secret_sauce", false},
{"problem_user", "secret_sauce", true}, //valid
{"performance_glitch_user", "secret_sauce", true},// valid
{"invalid_user", "invalid_pass", false},
{"", "secret_sauce", false},
{"standard_user", "", false}
};
}
@Test(dataProvider = "loginData")
public void testLoginWithMultipleCredentials(String username, String password, boolean
expectedSuccess) {
WebElement usernameField = driver.findElement(By.id("user-name"));
WebElement passwordField = driver.findElement(By.id("password"));
WebElement loginButton = driver.findElement(By.id("login-button"));
usernameField.clear();
usernameField.sendKeys(username);
passwordField.clear();
passwordField.sendKeys(password);
loginButton.click();
}
```

```
}
//6.


<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="All Parallel Tests" parallel="methods" thread-count="5">
<test name="All Test Classes">
<classes>
<class name="Testng.dataprovider"/>
<class name="Testng.dependent"/>
<class name="Testng.enableddisable"/>
<class name="Testng.loginpage"/>
<class name="Testng.multipleruns"/>
</classes>
</test>
</suite>
```