

Code

App

```
package JAVA_TESTING;

import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.TimeoutException;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
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;
import io.github.bonigarcia.wdm.WebDriverManager;

public class App {

    static WebDriver wd;

    @BeforeClass
    public void setUp() throws InterruptedException {
        WebDriverManager.chromedriver().setup();
        ChromeOptions option = new ChromeOptions();
        option.setBinary("C:\\Program Files\\BraveSoftware\\Brave-Browser\\Application\\brave.exe");
        wd = new ChromeDriver(option);

        // Navigate to the web application
        wd.get("https://www.saucedemo.com/");
        wd.manage().window().fullscreen();
        Thread.sleep(10000);
    }

    @Test(priority = 4)

    public void registrationTest() throws InterruptedException {
        try {
            wd.get("https://www.saucedemo.com/");
            WebElement usernameInput = wd.findElement(By.id("user-name"));
            usernameInput.sendKeys("nareshrao");

            WebElement passwordInput = wd.findElement(By.id("password"));
            passwordInput.sendKeys("nareshrao");

            WebElement submitButton = wd.findElement(By.id("login-button"));
            submitButton.click();
        }
    }
}
```

```

        wd.manage(). timeouts(). implicitlyWait(10, TimeUnit.SECONDS);

        // Add a condition to check if registration is successful
        WebElement successMessage =
wd.findElement(By.xpath("//div[@class='success-message']"));
        if (successMessage.isDisplayed()) {
            System.out.println("Registration Successful!");
        } else {
            throw new TimeoutException("Registration failed. Custom
Error: cannot register");
        }

    } catch (TimeoutException e) {
        System.out.println("Error Message: " + e.getMessage());
    } catch (Exception ex) {
        ex.printStackTrace();
    }

    Thread.sleep(10000);
}

@Test(priority = 1)

public void loginTest() throws InterruptedException {
    // Your login automation code here
    WebElement usernameInput = wd.findElement(By.id("user-name"));
    usernameInput.sendKeys("standard_user");

    WebElement passwordInput = wd.findElement(By.id("password"));
    passwordInput.sendKeys("secret_sauce");

    WebElement submitButton = wd.findElement(By.id("login-button"));
    submitButton.click();

    Thread.sleep(10000);
}

@Test(priority = 2)

public void searchAndAddToCartTest() throws InterruptedException {
    wd.manage(). window(). maximize();

    WebElement searchInput =
wd.findElement(By.xpath("//*[@id=\"item_4_title_link\"]/div"));
    searchInput.click();

    WebElement addToCart = wd.findElement(By.id("add-to-cart-sauce-labs-
backpack"));
    addToCart.click();

    WebElement showCart =
wd.findElement(By.className("shopping_cart_link"));
    showCart.click();

    Thread.sleep(10000);
}

```

```

    }

    @Test(priority = 3)

    public void invalidLoginTest() throws InterruptedException {
        wd.get("https://www.saucedemo.com/");
        wd.manage().window().maximize();

        WebElement usernameInput = wd.findElement(By.id("user-name"));
        WebElement passwordInput = wd.findElement(By.id("password"));
        WebElement submitButton = wd.findElement(By.id("login-button"));

        // Enter invalid username and password
        usernameInput.sendKeys("unknown_user");
        passwordInput.sendKeys("unknown_password");
        submitButton.click();

        // Wait for the error message to be visible (adjust the locator based
on your application)
        WebDriverWait wait = new WebDriverWait(wd, 10);
        WebElement errorMessage =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.cssSelector("div.
error-message-container h3")));

        // Check if the error message is displayed
        if (errorMessage.isDisplayed()) {
            // Print the error message to the console
            System.out.println("Invalid login error: " +
errorMessage.getText());
        } else {
            // If error message not found, print a generic message
            System.out.println("Invalid login attempt, but no error message
displayed.");
        }

        Thread.sleep(10000);
    }

    @AfterClass
    public void tearDown() {
        wd.quit();
    }
}

```

Pom

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>SPRING_STANDARD_TESTS</groupId>
  <artifactId>JAVA_TESTING</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>JAVA_TESTING</name>
  <url>http://maven.apache.org</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <dependencies>

    <dependency>
      <groupId>io.github.bonigarcia</groupId>
      <artifactId>webdrivermanager</artifactId>
      <version>5.6.2</version>
    </dependency>

    <dependency>
      <groupId>org.seleniumhq.selenium</groupId>
      <artifactId>selenium-java</artifactId>
      <version>3.141.59</version>
    </dependency>

  </dependencies>
</project>
```