

In the Page Object class, we should not include any validations.

Validations should always be part of the test case.

The method `getConfirmationMsg()` will only capture the text value and return it.

If the confirmation message is not displayed, it will throw an exception.

```
public WebDriver driver; // Declare this driver variable as a global class variable.
```

```
driver = new ChromeDriver(); // Initialize the ChromeDriver.
```

```
driver.manage().deleteAllCookies(); // Delete all cookies from the web page, if any are already stored.
```

We are using a **public** method because we are working with multiple packages. This allows the method to be accessed throughout the project, not just within a specific package.

We use **@BeforeClass** and **@AfterClass** because the two methods, **setup()** and **tearDown()**, should be executed only **once** before and after all the test methods in the class, respectively.

Dynamic Test Data:

- In Java, there is a predefined class called **RandomStringUtils**.
- It has a static method called **randomAlphabetic** with one parameter that specifies the number of characters in the string.

```
RandomStringUtils.randomAlphabetic(5); // Returns a random string with 5 characters
```

- You can call the method **randomString()** directly within the same class:

```
regpage.setEmail(randomString() + "@gmail.com");
```

- For **First Name** and **Last Name**, you can convert the random string to uppercase:

```
regpage.setFirstName(randomString().toUpperCase());
```

```
regpage.setLastName(randomString().toUpperCase());
```

- To generate numeric strings, use:

```
RandomStringUtils.randomNumeric(10); // Returns a random numeric string with 10 digits
```