Class Notes-23. _ day15

Test Case
open the chrome
go to qa1.vytrack
login as a sales manager
navigate to customers ->contacts
click on email mbrackstone9@example.com
verify that full name is Mariam Brackstone
verify that email is mbrackstone9@example.com
verify that phone number is +18982323434

TestBase vs BasePage

TestBase
we collected whatever common to all testcases here. Like before and after method
we created all the necessary object like driver, action,wait etc. and also reporting
setup. So we dont needt o copy/paste anymore. when we create the test we extend
TestBase

BasePage
we locate all the common elements and functionalites for all the pages/modules
here related to applicaition(vytrack). All other pages that have those functionlites will
extend the BasePage. it is application specific.

==
==
XML RUNNERS

what is xml file?
file type used to store and transfer data. It uses html like format, unlike to html it
does not have predefined tags.

Create testng xml runner file
1.Right click on the name of the project
2.New->File
3.Enter name: testng_runner.xml
4.OK

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd" >
this shows what kind of xml file is this, makes it xml runner for testng

How to run all test cases, classes inside the ine package ?

```
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Vytrack Automation" verbose="1">
<test name="Vytrack Smoke Test">
<packages>
<package name="com.cybertek.tests.day14_extent_reports"></package>
```

```
</packages>
</test>
</suite>
```

How to run multiple package ?
```
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Vytrack Automation" verbose="1">
<test name="Vytrack Smoke Test">
<packages>
<package name="com.cybertek.tests.day14_extent_reports"></package>
<package name="com.cybertek.tests.day11_webtables"></package>
</packages>
</test>
</suite>
```

Note: xml runner looks for @Test annotatuin, if your test case under main method, it will NOT run.
How to run everthing(all packages) --BATCH RUN--
```
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Vytrack Automation" verbose="1">
<test name="Vytrack Smoke Test">
<packages>
<package name=".*"></package>
</packages>
</test>
</suite>
```

How to run specfic class ?

```
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Vytrack Automation" verbose="1">
<test name="Vytrack Smoke Test">
<classes>
<class
name="com.cybertek.tests.day14_extent_reports.NegativeLoginTestWithReport"></class>
</classes>
</test>
</suite>
```

NOTE: verbose is controlling log reports starting from 1, the more you add number it will give more details.

How to run more than one classses (multiple classes)?
```
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Vytrack Automation" verbose="1">
<test name="Vytrack Smoke Test">
<classes>
<class
name="com.cybertek.tests.day14_extent_reports.NegativeLoginTestWithReport"><
```

```
/class>
<class name="com.cybertek.tests.day13_pom.PageSubtitleTest"></class>
</classes>
</test>
</suite>
```

How to run specific test case from the class ?
we use include tags.
Include method is useful when you have multiple test cases in a class and you only want to run one or a few of them.

```
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Vytrack Automation" verbose="1">
<test name="Vytrack Smoke Test">
<classes>
<class name="com.cybertek.tests.day11_webtables.WebTablesExample">
<methods>
<include name="printTableSize"></include>
</methods>
</class>
</classes>
</test>
</suite>
```

Exclude method (sadece bir test case I cikariyorsun , digger test ler calisiyor sinifta)
Exclude methods is useful when you want to omit (cikarmak) one testcase from class and run rest of the test cases.

```
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Vytrack Automation" verbose="1">
<test name="Vytrack Smoke Test">


<classes>
<class name="com.cybertek.tests.day11_webtables.WebTablesExample">
<methods>
<exclude name="getAllCellInOneRow"></exclude>.          ***  Or include kullan
</methods>
</class>
</classes>
</test>
</suite>
```