POM D.W que POM

1. What is POM?
2. Explain architecture of POM
3. what is page chaining model
4. use of PageFactory API in selenium
5. Explain framework.

POM → Page Object Model → selet

Object ⇒ element

POM ⇒ ① organizing the page object or page elements.

Normally, organize using Java programs

generally,
① find elements or operat
② perform action on element

① Identification
② Operation / action

In real time, we need to automate 100 or more no. of pages.

driver.findElement(By.name("xxx")).sendKeys(); ⇒ In this
1 part (identification) 2nd part (operational method)

model, we are writing Identificatn & operatn in one line, which is not recommend for to automate two pages, so we go for POM.

So, In POM, we divide ~~one~~ whole application statement in 2 different part:-
class i.e. class 1 ⇒ Identification
class 2 ⇒ operational method

for eg. we have one application which have 5 diff. pages
eg. login, home, customer, transaction screen.

For, for each page having diff. elements in web

- login - 3
- home - 5
- customer - 20
- transactn - 30

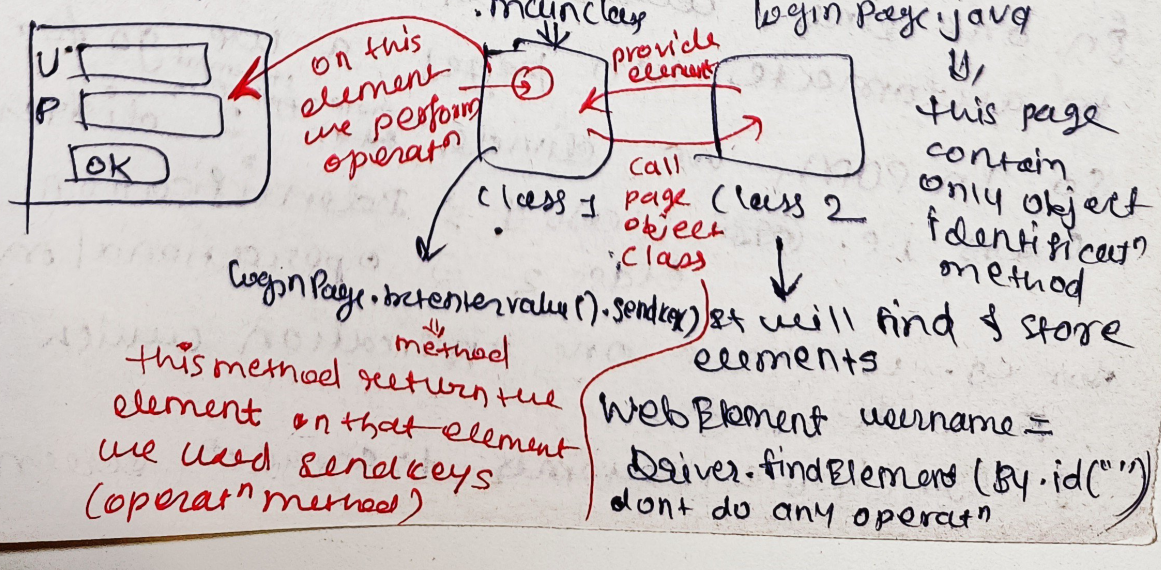
→ this class contain only identificatn method not operational method

what we need to do in real time for every page you we will create one class, the class contains only identification method

then we create one main class. or test class ⇒ from this particular class, we will do operations & to perform ~~elem~~ operation we require element, so we will call that classes (login, home etc) to find interact with the elements.

For eg.

I have login screen



⇒ What is POM?

- Maintaining or ~~defining~~ managing all the page objects in a proper way on different files is a POM.

⇒ Why POM?

If I write everything identification & operational method in a same class" itself if we have huge statement or something is repeating multiple times then we need to go everywhere & change so tomorrow some properties are dynamically changing or they are not constant values so what you can do everytime whenever the change is happen you need to go there & change the particular statement, this will happening in real time. If there is ~~pro~~ complex project then we continue keep on will face a lot of issues, so the solution is we are dividing object identification method in one class & operational method in class

POM - by Naveen

⇒ It is approach. It is also called different design pattern.

→ Integrate with data driven framework.
(testNG)

⇒ for each page available in webpage we need to create seperate class for it.

if 100 page → we will create 100 Java class

web elements / web objects ⇒ wherever seeing on web page like header, footer, links, textbox, buttons, text

methods ⇒ Get the title of the page
login functionality → click on signin, register

clickOnSingleLink(), fillRegform(), checkLogo(), footerLinks()

1st layer will be ⇒ page layer

→ define all objects & actions for each, page. this are called my page
& every
libraries.

⇒ web element / web objects are also called as object Repository.

Object Repository means collection of all web elements or web objects or objects

2nd layer ⇒ Test layer

we have to create test layer.

→ If I want to write a test cases for login page, then I will create seperate Java class for that.

→ this test layer will be written with the help of testNG

→ Somewhere betn 1st layer & 2nd layer we need to create one base class which will parent class of all class

→ Test Base class contains (3rd layer), all prerequisite are require

for eg. ① create driver ② properties ③ initialization ④ maximize window method (maximizeWindow()) ⑤ pageloadTimerPut() ⑥ implicitWait() ⑦ deleteAllCookies() ⑧ get(url) ⇒ All this thing define in base class

→ common properties which ~~are using~~ will use in ~~base classes~~ pages or in test classes that will define in base class

Why base class ⇒

↳ To avoid unnecessary repeatation of code.

so, then use concept of Inheritance. (parent & child relationship) then access parent class directly.

→ so, it will reduce code & it will be systematic.

3rd layer ⇒ config. properties / Environment variables

like ~~like Environment Variables~~ like url, username, password, browser property (which browser use) common properties .. etc.

⇒ Repository means collection of all web elements or web objects or objects.

2nd layer ⇒

We have to create

5th layer ⇒ testdata.xlsx → to maintain data

where we will store data,

→ we will store our data in excel file, which will in the form of rows & colm

→ Read excel file by using Apache POI file. can create diff. sheets for diff. modules.

test data package

6th layer ⇒ Utility → create some utility

utility like screenshots, if you want to send mail some common utility. Screenshot(), sendMail(), commonutil() → this are Generic functions.

7th layer ⇒ Reporting (Test Report : pass/fail)

it can be anything like .HTML report, TestNG report, xml report, test Reports. (How many test case passed, fail) we can use Extent Report.

⇒ Repository means collection of all web elements or web objects or objects.

2nd layer ⇒

We have to create

5th layer ⇒ testdata.xlsx → to maintain data where we will store data,

→ we will store our data in excel file, which will in the form of rows & col^m

→ Read excel file by using Apache POI file. can create diff. sheets for diff. modules.

test data package

6th layer ⇒ utility → create some utility utility like screenshots, if you want to send mail some common utility. Screenshot(), sendMail(), commonutil() → this are Generic functions.

7th layer ⇒ Reporting (Test Report : pass/fail) it can be anything like .HTML report, TestNG report, xml report, test Reports. (How many test case passed, fail) we can use Extent Report.

→ Each & everything will present in particular package.

eg. com.qa.pages, com.qa.tests, com.qa.base, config pkg, testdata data pkg, Reporting folder (OIP folder), com.qa.config, com.qa.testdata, com.qa.util, com.qa.Test-output/reports

⇒ Home page is a landing page of login page.

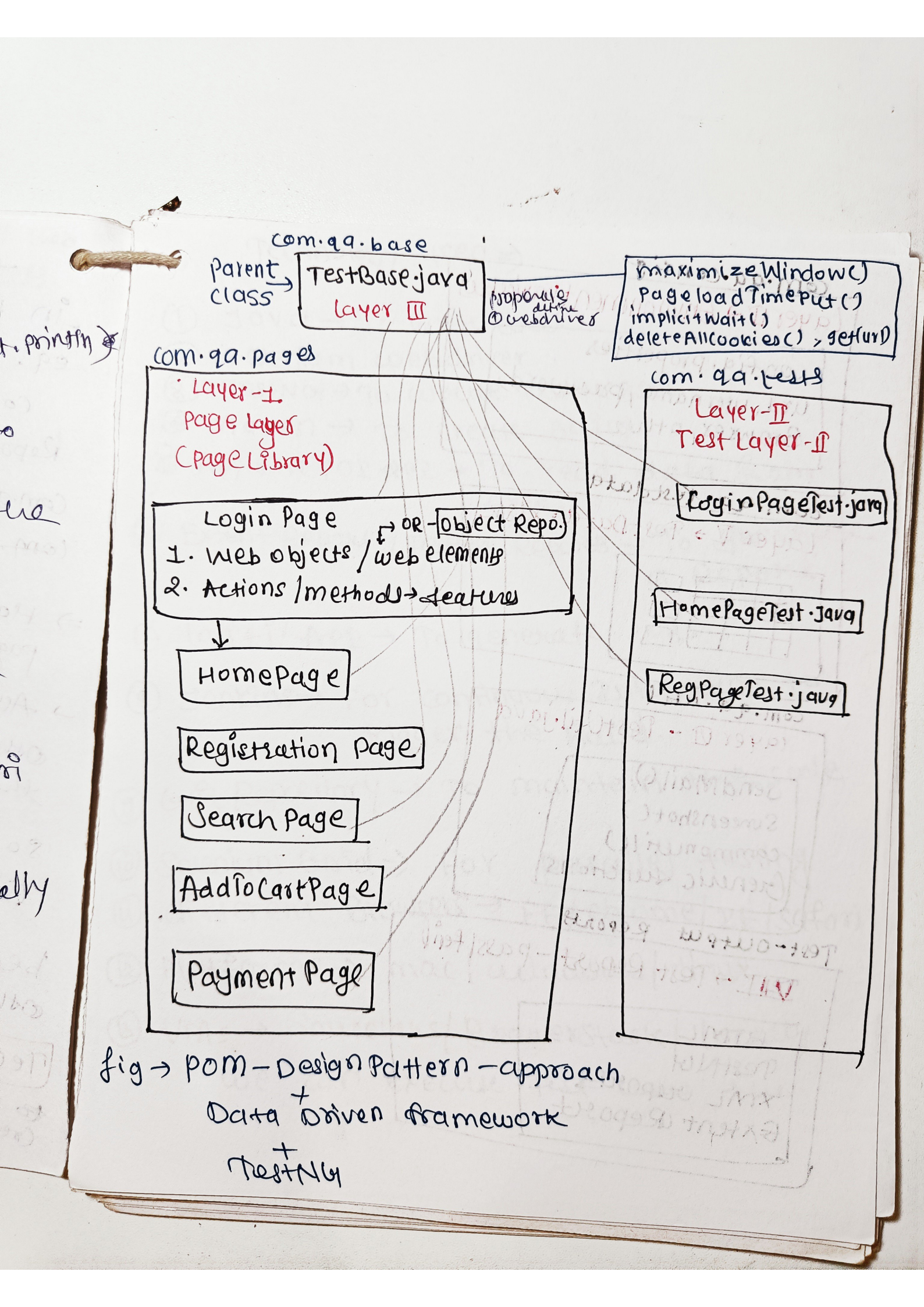
→ All pages are interconnected with other, can do diff navigation on this class.

So, page Object model is also called Page chaining model.

because it is interconnected to each other, like chain.

Technology used → Java, to write code selenium webdriver, to write test case using testNG, for creating build we are using maven, Appache POI

API to read data from excel.
extent Report / TestNG report to generate Reports, log4j / API to generate logs (instead of system.out.println we will be using log4j)
Jenkins → for continuous integration to trigger the Build.
Git Repository → to maintain the code.
(for pushing code once in & checkout point of view)
Selenium Grid ⇒ for parallel testing
diff Browsers ⇒ FF / chrome / IE / safari
platforms ⇒ MAC / Windows / Linux
VMs / SauceLabs / BrowserStack / Locally
⇒ can execute test cases on here


com.qa.base
Parent class
TestBase.java
Layer III
properties define
driver
maximizeWindow()
PageloadTimePut()
implicitWait()
deleteAllCookies() , getUrl()
com.qa.pages
Layer-I
Page Layer
(Page Library)
Login Page
OR - Object Repo.
1. Web objects / web elements
2. Actions / methods → features
HomePage
Registration Page
Search Page
AddToCartPage
Payment Page
com.qa.tests
Layer-II
Test Layer-II
LoginPageTest.java
HomePageTest.java
RegPageTest.java
fig → POM - Design Pattern - approach
+
Data Driven framework
+
TestNG

com.qa.config

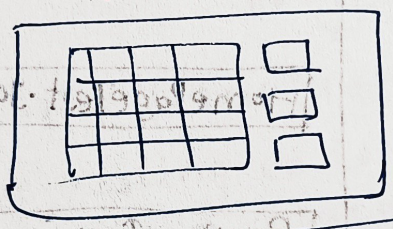
Layer III → Environment variables

config.properties

url, username, password,
Browser, other

com.qa.testdata

Layer V → TestData.xlsx



com.qa.util

layer VI - TestUtil.java

sendMail()
Screenshot()
commonUtil()
Generic functions

Test-Output Reports

~~Test~~ III → Test Report - pass/fail

HTML
TestNG
XML reports
Extent Report

Technology used ⇒

① Java → To write code
② selenium webdriver
③ TestNG → To write test cases
④ maven → To create Build
⑤ Appache POI API → To read data from excel
⑥ Extent Reports / TestNG Reports → To generate a Report
⑦ log4j / API → To generate logs
⑧ Jenkins → For continuous Integration to trigger the Build.
⑨ Git Repository → To maintain the code
⑩ Selenium Grid → for parallel testing
⑪ different Browsers → FF / chrome / IE / safari
⑫ Platforms → mac / windows / linux
⑬ VMs → Sourcelabs / BrowserStack / locally we can execute test cases here.

Explain POM in Detail

① POM is approach or different design pattern

I integrate it with data driven framework & testNG.

② So, the first component of is page layer, where defines all objects & web elements & web objects for each and every page that is called the object repository. ~~or~~ ~~use~~ So, the object repository is the collection of all web elements or web objects.

② Now the second component of my project is test layer, where I have created test cases for all the pages which define in page layer, for that I created Java class for that.

& wrote the test cases with the help of testNG

③ The 3rd component is Base class which is the parent class of all class present in project.
It is implemented by using inheritance
The all common properties which will use in page or in test class that will define in base class.
eg. (initialization, maximize window method, pageloadtimeout, implicit wait(), deleteAll-cookie, geturl)
So, that it will reduce code & it will be systematic.

④ The 4th component is Environment variables or config properties like url, username, password, browser property which is common properties

⑤ The 5th component is test-data.xlsx to maintain the data, we will store our data in Excel, which will in the form of rows & col^m
Read Excel file by using Apache POI file.
can create diff. sheets for diff. modules.

⑥ The 6th component is created some Utility → like Screenshots, that will contain some common utilities.

⑦ 7th component is → Reporting
The test reports like pass / fail
It can be anything like HTML reports, TestNG report, xml report, test reports or extent Report.
But I used . . . . in my project.

This → Each & everything will present in particular package.

→ All pages are interconnected with other, we can do diff navigation on this class So, it is also called as page chaining model.

Practical

This is kind of hybrid approach
it's combination of Page object model
~~POM~~ + data driven approach.

→ pom.xml file is the heart of project
we have to add all the dependencies
over there (what diff libraries are you
using) eg. APache POI API, extent report
API, Selenium webdriver API.
(No need to download, give dependencies)

STEP(1) =>
i) dependencies for selenium
→ google → webdriver maven
↳ copy paste under <dependency> </>
→ we can change version
The Build will be taken care by
maven → So maven is Build Automat? tool.

ii) add testNG
write → testng maven → copy & paste
on
google

iii) APache POI API → using data driven point of view
reading a data from excel
sheet

type ⇒ cntr+s ⇒ it will start downloading
all files

I.w qu ⇒ what are your folder structure in your framework.

i) How many packages you have created?
ii) where exactly no. of files are available?
iii) what are diff source folder you have created,
iv) How exactly the data is available?
v) where exactly the data is available?

framework always designed step by step

project tips ⇒ ① decide approach
② define dependencies you need
③ define folder structure

imp folder

→ src/main/java → whatever the utility, page classes, libraries entire thing created in src folder

→ src/test/java → only and only to written a test cases. test cases which are written in testng

Creating package wise component

⇓

Step (2)→(1) create package under src/main/java

com.crm.qa.pages (first letter)

(2) create package under src/test/java

com.crm.qa.testcases

Step (3) ⇒ create base class → this is utility

→ in src/main/java ⇒ make package

✓ com.crm.qa.base

⇒ create Environment variable →

config.properties

• create another package

✓ com.crm.qa.config

⇒ create Data point of package

→ src/main/java →

✓ com.crm.qa.testdata

⇒ Utilities

✓ → com.com.qa.util (common utilities will be there)

Step (4) ⇒ See ~~Pages~~ website ⇒ ~~how~~

page → how many pages are there & how many pages are automated

① define configuration

com.crm.qa.config → create a file with config.properties name

url →
username →
password →
which browser

don't create test data in config file

properties file are only for global Environment

Step ⑤ ① create login page class in com.crm.qa.pages

LoginPage → don't select main class

② java class for sign up page

create class for each and every page → rule of POM

for every page → we have to create a seperate class.

100 pages → 100 java classes

① ContactsPage.java
② DealsPage.java
③ HomePage.java
④ LoginPage.java
⑤ SignUpPage.java

Step ⑥ ⇒ Create base class in base package

properties → variables + methods

we are inheriting some properties from base class to other class.

write code in base class like

*) instantiate browser, window minimize

① → driver.manage().window().maximize();
driver.manage().deleteAllCookies();
driver.manage().timeouts().pageLoad
(20, TimeUnit.SECONDS);
driver.manage().timeouts().implicitlyWait(10, T)

→ In future time like 20, 30 can inc., then again you have the change inside the script, so better to do that we create one util class inside util package

testUtil class → Create ② method

→ In login page/we have to define object ~~factory~~ Repository, page factory of login page.

Q10) How will you initialize ~~method for~~ page factory?

There is method called PageFactory.initElements(driver, this)

↑ current test object

↑ all web elements will initialize with driver

① define web elements } in login page
② define methods

↳ features/actions

Step ⑥ create LoginPageTest class in com.crm.qa.testcases

Technology used

1. Selenium webdriver (open source)
2. JDK (Java development kit)
3. TestNG (Test Unit framework)
4. Log4j (logging API)
5. Maven (Build Automation tool)
6. Apache POI API (Read-write utilities for excel - Test Data Handling)
   Eclipse (Java Editor)
7. Browser - Google Chrome / FF

Automation framework Architecture

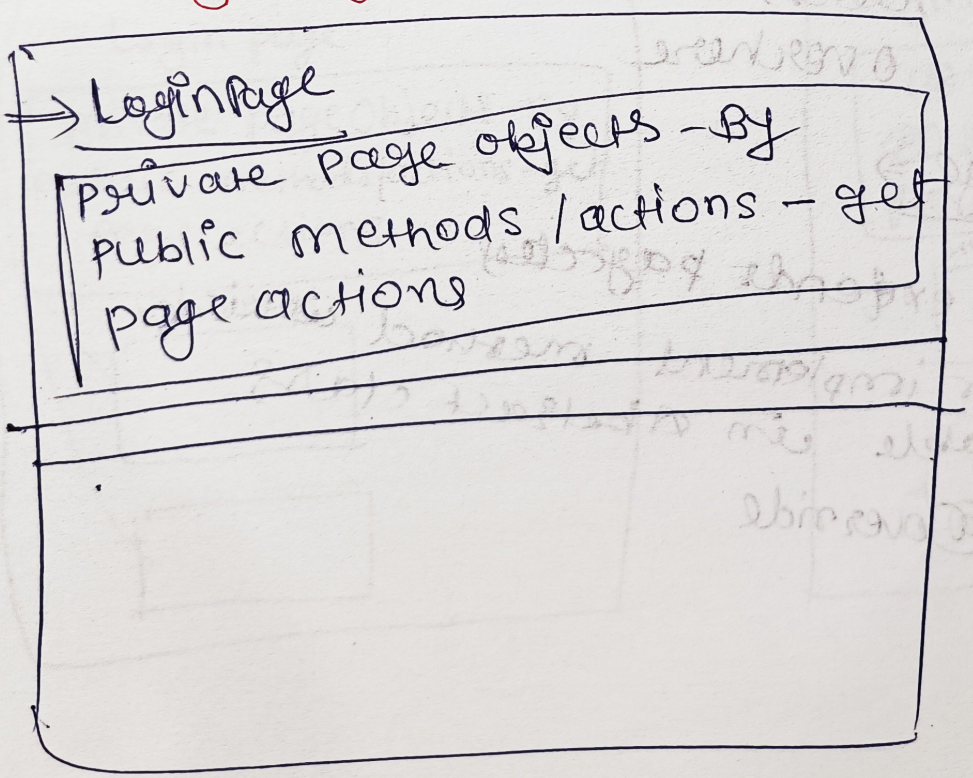
1. POM design - Page factory API of WebDriver
2. maven (BAT)
3. Test Libraries for diff. UI pages
4. Test Utilities for diff. generic fun^{n}
5. Report - Dashboard (Pass/fail) by using Extent Report.
6. API Jenkins - continuous integrat^{n} too)
7. Git Hub Rep (code versioning tool)

OOPs Concepts

Page objects by using By locator not used page factory.

Abstract method →
A method which is having one abstract keyword & does not have any method body that is called abstract method.

page layer →



② Abstract Page class

- Can create abstract or non abstract class in page class.
- prototype methods →
  - only rules will define
  - what are the diff methods should have every page
- getInstance(Page class){
  return object of page class
  } ⇒ return object of specific class that we are passing everywhere

m1()
m2()
m3()

③ Base page ⇒

Base page extends page class
→ define or implement method which is available in Abstract class
→ @Override

m1()
m2()
m3()

Test layer

m1()
m2()
m3()

Abstract Page Class

Prototypes methods

getInstance(page class){
  return object of page class
}

BasePage extends Page

m1(){} @override
m2(){}

BaseTest

@BM → driver initializat^
@AM → driver.quit()

Login page

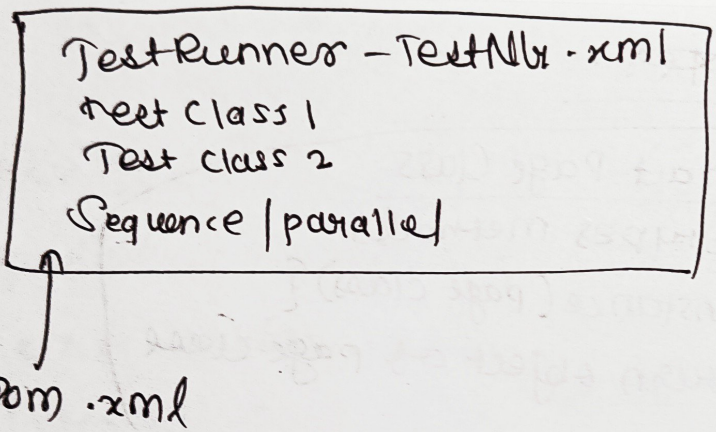
private pageObjects - By
public methods/actions - get page actions

Home page

Login pageTest

@Test
@Steps
Assertions

Home page


TestRunner - TestNG.xml
Test Class 1
Test Class 2
Sequence | parallel
pom.xml

Test layer

① **Base Test** → maintain drivers

① @BM → driver initialization

② @AM → driver.quit()

② **LoginPageTest**

@Test
steps
Assertions

concentrate only on writting test cases

There will be integration betn pageclass & test class

③ TestRunner – TestNG.xml

TestClass1
TestClass2
sequence/parallel

we will give the path of testng.xml file which will run your test cases from here

pom.xml (contains all dependencies or plugins)

OOPS

→ OOP – Inheritance
→ Method Overloading
→ Method Overriding
→ Encapsulation – private | public
→ Java Generics
→ Wrapper methods – Generic Page Actions
→ Abstract class (create abstract class constructor)
→ Object Creation Strategy
→ Constructors

Wrapper classes → for data conversion

Java Generics → type Safety

Extent Report Generation

(1) add library in pom.xml file

(2) add dependency for extent report

or

You can download jar files from google.

use → It gives very good look & feel for dashboard.

(2) → create package under src/main/java

Com.qa.ExtentReportListner

→ use listner

→ There is standard template provide we will add for extent Report

→ with the help of testNG listner they have created ExtentReporterNG class

→ we can give any name

Listner → We have to add listner entry into testNG.xml.

step (3)

purpose → Listner will listen each and every activity of your execution and then on the basis of those activities it will generate the report at the end.

testng.xml ⇒ immediate after <suite> tag

<listeners>

~~<listener>~~

<listener ~~class~~ class name = " ~~exact~~ give path with / package name if it is available

"exactpkg.classname"

<class name = "com.crm.qa.LoginPageTest">

: classes "

: :

→ It will listen all this classes, what are activities are going on at the time of execution, it will listen & then it will good to go

→ due to that ~~report~~ listener entry it will create HTML report, report name will be given in listener class which will available in Output Directory.

→ then run testng.xml file

→ to see the report just refresh the project & then test-output → extent.html → right-click → properties → copy path → paste in browser

In extent.html

- → test tap
- → Defect tap
- → complete Automation tab

→ It is user friendly
→ we can search ~~user~~ test cases
→ we can refresh
→ we can give filter

① Extent reports is an open source library that generates customizable HTML reports.

Selenium Reporting tools → Extent Report

② It is an open source reporting library used in selenium test automation. It provides more extensive & insightful perspective on the execution of your automation scripts.

Adv ① Extent Reports are more customizable than others.

② It can be easily integrated with frameworks like JUnit, NUnit & TestNG

③ Extent API can produce more interactive reports, a dashboard view, graphical view, capture screenshots at every test step, & emailable reports.

How to Integrate with Jenkins

i) install and configure Jenkins
ii) Setup a maven project
iii) Run Automation Test Suits from Jenkins

→ Jenkins is available in war file
  i) download Jenkins war files (74mb)
→ go to download folder → copy it → & paste in seperate folder created as name Jenkins. (In C or d drive)
→ Run that file through cmd prompt
  cmd→
→ go to that folder where Jenkin is available.

cmd⇒ java -jar jenkins.war ↵
  # It will start initialization process for Jenkins. It will create all directory for Jenkins in your laptop.

⇒ if you see * * like this
  Jenkins initial setup is required
An admin user has been created & a Password generated.

→ Jenkins always Run ~~[illegible]~~ 8080

By default it Runs on ~~[illegible]~~ 8080 port

→ goto browser → type localhost:8080 ← here

Copy password from cmd & paste

→ Install suggested plugin

→ create first Admin User

username –
password –
Confirm password –
full name –
E-mail address –

→ Jenkins is Ready

→ we can create multiple people.

→ click on new item

↓ Not available maven plugins then do this

We have created our project on maven so, we require maven plugin so,

go → manage Jenkins → manage plugins → Available → search for maven → Select all maven related plugin (select it) → ↓ download, restart → click on

→ Restart Jenkins
→ we can see logs on cmd prompt
So don't close cmd prompt
otherwise Jenkins will stop.

→ then login

→ Jenkin → new item → Enter an item Name → FreeCRMTestRunner → select maven project ⇒ click on ok

→ maven project Name →
Descriptn → free CRM Test Lab: test automation suit for free CRM App
(don't do anything for maven info plugin configuration)

→ If you want to add some other repository then you have to add some plugins for that repository.

→ click on tool configuration to add your project → add maven name → apply & save it

Root POM ⇒ go to in your project → right click on pom.xml → properties → select path → paste here

goals & option ⇒ clean install → install all artifacts & dependencies then will execute your test cases
add result plugin → testNG plugin

→ see logs in cmd prompt
Jenkins is restarting

→ configure → see → publish testng results

testNG xml report pattern → **/testng-results.xml

** ⇒ whatever the location directories
** will pick that particular directory,
it will search testng-results.xml file
from workspace from any directory which is available
in your workspace.

↑ Jenkins will create its own workspace
internally.
one copy of ~~jenkin~~ will create for jenkin
also. workspace

In that workspace, search particular
course is exactly the testNG-result.xml
file is available. it will read that
xml file & it will show you the
report over there. → Apply

⇒ trigger the Build
click on Build Now

Parsing the pom.xml file & it will see is there anything to execute or not.

⇒ then one by one test cases will execute

⇒ In cmd prompt for → Jenkins execution or for Jenkins setup configuration O/P will be shown here.

⇒ But In Jenkins, we can see the entire execution, you can monitor All the test cases will execute one by one.

⇒ So, ~~we can directly~~ execution through Jenkins.

So, we can install Jenkins on server & then you can access that from their m/c, you can directly execute build or automate from your laptop without disturbing your work.

⇒ We can see complete error log here.

you know now

1) How to setup Jenkins
2) How to download Jenkins warfile
3) How to Run your war file
4) How to setup the project & all the plugins like maven, testing plugin we have to download.
5) How to Configure the complete job over here. & how to execute it & then how to see the result respective build.

But for manage plugin like junit, maven, gradle or cucumber if you are using then you have to download plugin.

run test cases from Jenkins ⇒ give pom.xml file to them.

Selenium + Jenkins + GIT integration

Run your Test cases from GIT Hub using Jenkins

→ Github & Jenkins integration for Selenium Project
→ Git Repo configuration in Jenkins Jobs
→ Continuous integration with selenium webdriver
→ How to publish testng reports
→ How to publish extent reports
→ Jenkins selenium webdriver testng
→ How to run testng.xml in jenkins
→ Jenkins Selenium plugin
→ How to configure testng.xml in jenkins

Jenkins integration with GIT (SCM)

→ Executing our test cases directly from Jenkins not from eclipse we have given pom.xml file over there..

→ But pom.xml file is coming from local system/mic.

But what if check in code in git repository & then directly execute from git repo.

→ integrate git with Jenkins

→ Jenkin should pick the code directly from git repo. & then it should execute from git only.

① → git status ⇒ to check the status

② → git add. ⇒ it will add all the complete repository folder structure

→ git status ⇒ seen green color means added into the bucket.

commit → (3) git commit -m "added minor changes"

push in repo → (4) git push origin master

username-

for reset password
→ ① git push origin master
② username

after push up code in github then start Jenkins

cmd ⇒ cd.. go to that folder where Jenkins is present
⇒ cd Jenkins/
⇒ ls
⇒ java -jar jenkins.war # start Jenkins

google chrome ⇒ localhost8080
↳ if loginpage is there that means your jenkin is working fine

Now → Source code

Git ⇒ give url.

by default → master branch

if other branch ⇒ */ branch name

if I jenkins से run करना है तो clone बनाना पड़ता है।

integrate extent report ⇒

from ~~result~~ test-output folder.

Configure → में Set path

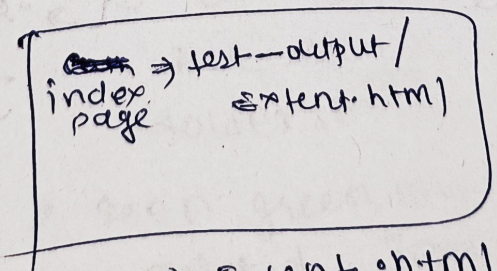
⇓

अगर इसमें नहीं मिला कुछ तो download

Plugin

publish build Actions

→ Reports → add कर देना [configure से आएगा]



index page ⇒ Extent.html

Now → Source code

Git ⇒ give url.

by default → master branch

if other branch ⇒ */ branch name

if jenkins से run करना है तो clone बनाना पड़ता है।

integrate extent report ⇒

from ~~result~~ test-output folder.

Configure → में set path

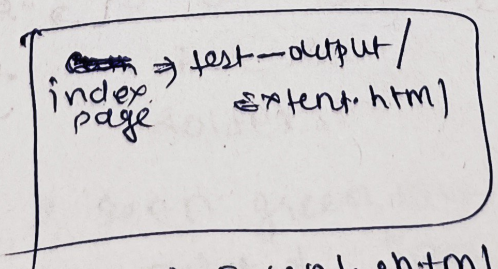
↓

अगर इसमें नहीं मिला कुछ तो download

plugin

publish build Actions

→ Reports → add करेंगा [configure से आएगा]



index page ⇒ Extent.html

for good graphical UI of extent file
→ Jenkin → manage jenkin → script console
→ set one property

System.setProperty("hudson.model.DirectoryBrowser
Support.CSP","")

Click on Run

O/P
Result
→ Result (को शॉट लगाना चाहिए)

How to add Screenshot in Extent Report

1. Create maven project
2. add dependency of extent report in pom.xml
3. class ~~TC211~~

⇒ class }

where the screenshot will generate

① // define all the extent parameter
@ before test
@ After test
// ~~Extent~~ flush close.

② // To take screenshot
@ before method
~~setup~~
@ After method
logic

pick only those test cases which got failed

So hence one Listener ITestListener

· ITestResult → coming from testNG

whatever the results are coming (100 test cases
~~pass~~ whether the count what test passed
or failed or skiped it will store in
ITestResult object

ITestResult result;
result.getName() → print errorname
result.getThrowable() → print descriptn of Screenshot

screen shot path is available

String screenShotPath = Free(RMTest.getScreen(driver, result.getName));

attach this Screenshot with extent test

extentTest.log(

How to generate log files in Selenium

→ How to generate log files in Selenium Webdriver
→ Selenium log file
→ log4j properties in Selenium webdriver
→ Download log4j jar for Selenium
→ log4j jar for selenium webdriver
→ Selenium webdriver console log

---

① create maven project
② add dependencies for selenium & testng
③ To generate log → add one more dependency called log4j dependency
download → log4j maven → copy dependency & added on there.
④ Create package → src/test/java
→ com.tests
→ class (LoginTest)

① Download latest log4j distribution
② Add log4j jar to library
③ create log4j configuration
④ create logger
⑤ put logger statement into your code.

I.W qus on log4j

① what is log?

log is To capture the information what is going on at a time of ~~exe~~ program execution.

when you execute your program, JVM java is also executing something, testng also executing something, so lot of API working together doing their respective work.

you went to know for each & every step, what are diff. execut°, what are diff. calls in background going on.

So you can understand the execut° properly that what exactly is going on.

lets see your selenium code is running executing some program or on some ~~more~~ remote m/c. everytime you cannot see the what is going on remote m/c everytime you will go to remote m/c, & then login into the system & check what is going on

No. you can monitor logs from directly your local system also that what is going on inside the log.

So, log is very imp in terms of debugging of code. what are the diff failures at run time, in terms of to check the exceptions in any run time exceptions or any exceptions are coming, what is reason behind that or file is missing or some jar is missing easily you can trace it from logs.

So, Log is capturing info or activity at the time of program execution

Que types of log?

① info
② warn
③ error
④ fatal

} ⇒ 4 levels of log

Que ⇒ How to generate a logs?

To generat a logs in java we used

Appache Log4j API.

Ques ⇒ what is mechanism, How it works?

It reads log4j configuration from log4j.properties files.
log4j.properties file is most imp file to generate a log.

Ques ⇒ where to create this file?

We have create this log4j properties file inside the resources folder
source

Right click on project → create source folder →
src/main/resources
↳ go to file (create file)
↳ log4j.properties
↳ compulsory to write this name only

properties means key & value format

In this file, we have to define properties on some key & value format.
Here are diff. levels are present so 1'st level is root category y debug inside console mode as well as, I want to generate a log in particular file also

↳ log4j.rootCategory=debug, console, file

Appender which writes to a file

① consoleAppender → (Appender name for console)
this appender which writes to console.

② patternlayout → we can generate, pattern like
log on
mm-dd-yyyy date wise, no. of ms, sec.
standard configuration

③ Appender which writes to a file
→ RollingFileAppender (appender name for file)

→ Append=false ⇒ just remove previous logs (just overwrite)
Append=true ⇒ Append with previous logs
don't override page logs)

you need just SSH into that m/c, you just need remote login it
into that m/c
By using SSH, we can see logs in
diff. m/c

→ go to director in console →
maven clean install

log.info() → to print

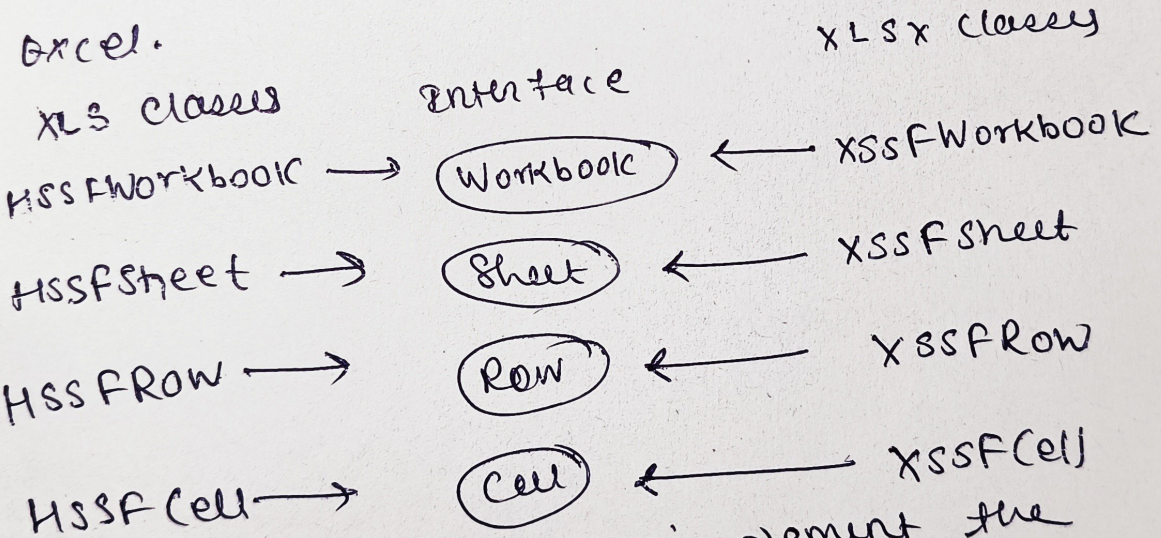
To show some warnings or alerts use

* log.warn( ) → to generate warning msg
* log.fatal( ) → print error msg
* log.debug( ) → print debug msg
* log.info( ) → entering applicatn URL.

Apache POI in Selenium

→ It is widely used API for selenium data driven testing.

→ To read & write excel file in Java, Apache provides a very famous library POI. This library is capable enough to read & write both XLS & XLSX file format of excel.



The two classes which implement the "workbook" interface are given below.-

① HSSFWorkbook → methods of this class are used to read or write data to microsoft excel file in .xls format.

② XSSFWorkbook - methods of this class are used to read/write data to microsoft excel & OpenOffice xml files in .xls or .xlsx format.