<https://sites.google.com/site/httpwwwseleniumhqorg/>

read from above website about selenium

**Selenium is a suite of tools** to automate web browsers across many platforms.

## Which part of Selenium is appropriate for me?

|  |  |
| --- | --- |
| Selenium WebDriver [Selenium LogoSelenium Grid Logo](http://www.seleniumhq.org/projects/webdriver/)  If you want to   * create robust, browser-based regression automation suites and tests * scale and distribute scripts across many environments   Then you want to use [Selenium WebDriver](http://www.seleniumhq.org/projects/webdriver/); a collection of language specific bindings to drive a browser -- the way it is meant to be driven.  Selenium WebDriver is the successor of [Selenium Remote Control](http://www.seleniumhq.org/projects/remote-control/) which has been officially deprecated. The Selenium Server (used by both WebDriver and Remote Control) now also includes built-in grid capabilities. | Selenium IDE [Selenium IDE Logo](http://www.seleniumhq.org/projects/ide/)  If you want to   * create quick bug reproduction scripts * create scripts to aid in automation-aided exploratory testing   Then you want to use [Selenium IDE](http://www.seleniumhq.org/projects/ide/); a Firefox add-on that will do simple record-and-playback of interactions with the browser |

**Introduction:**  
**Selenium** is an open source automation testing tool.  
It is used exclusively for web based applications  
 You can work on multiple operating systems using selenium

Platforms Supported by Selenium

* Windows
* OS X
* Linux
* Solaris

following languages are used with selenium

* - Java  
  - C#  
  - Ruby  
  - Python  
  - PHP  
  - Perl

Selenium Browsers Support:

* Internet Explorer
* Firefox
* Chrome
* Safari

\*\*Selenium Features\*\*

Selenium is open Source Automation Testing tool

It is exclusively for Web Based applications.

Selenium supports multiple browsers -   
Chrome, Firefox, Internet Explorer, Safari

Selenium works with Multiple Platforms   
Windows, Apple OS X, Linux

Selenium can be coded in multiple languages -   
 Java, C#, Python, Javascript, Python, php,Ruby

Difference between Selenium and Webdriver?

Misconceptions

|  |
| --- |
| 1. Classes should not have spaces- Compound classes cannot be accepted |
| 1. Multipl values - Selenium identifies the first one- Scans from top left |
| 1. Double quotes inside double quotes are not accepted |
| 1. Xpath/CSS can be defined in n number of ways |
| 1. Rightclick copy on blue highlighted html code to generate xpath |
| 1. Firepath depreciated from firefox- |
| 1. when xpath starts with html-Not reliable- Switch browser to get another one |
| 1. //tagName[@attribute='value'] - xpath syntax |
| 1. tagName[attribute='value'] -CSS tagName#id- CSS tagname.classname- CSS |
| 1. //tagName[contains(@attribute,'value')] - xpath regular expression |
| 1. tagName[Atrribute\*='value'] - Css regular expression       Selenium WebDriver Architechue Simplified:     * After you trigger the Test, complete Selenium code (Client) which we have written will be converted to Json format * Generated Json is sent to Browser Driver (Server) through http Protocol   Note: Each browser contains a separate browser driver   * . Browser drivers communicate with its respective browser and executes the commands by interpreting Json which It received on the browser. * Browser Driver receives responses back from the browser and it sends Json response back to Client. |
|  |

1)For java

<https://www.oracle.com/java/technologies/javase-downloads.html>

download only jdk

need to provide java path for system variables as selenium don’t exactly where is java it will directly go to system variables and search for java and then execute java exe files

JAVAHOME - C:\Program Files\Java\jdk-14.0.1

Give bin path

Path - C:\Program Files\Java\jdk-14.0.1\bin

2) eclipse

<https://www.eclipse.org/downloads/>

click on download packages and download

### [Eclipse IDE for Java Developers](https://www.eclipse.org/downloads/packages/release/2020-06/r/eclipse-ide-java-developers)

C:\Users\shmounik\eclipse\jee-2020-03\eclipse

-🡪 go to above path and open eclipse exe file and click on work bench

Java proj is treated as test suits and java classes are treated as testcases

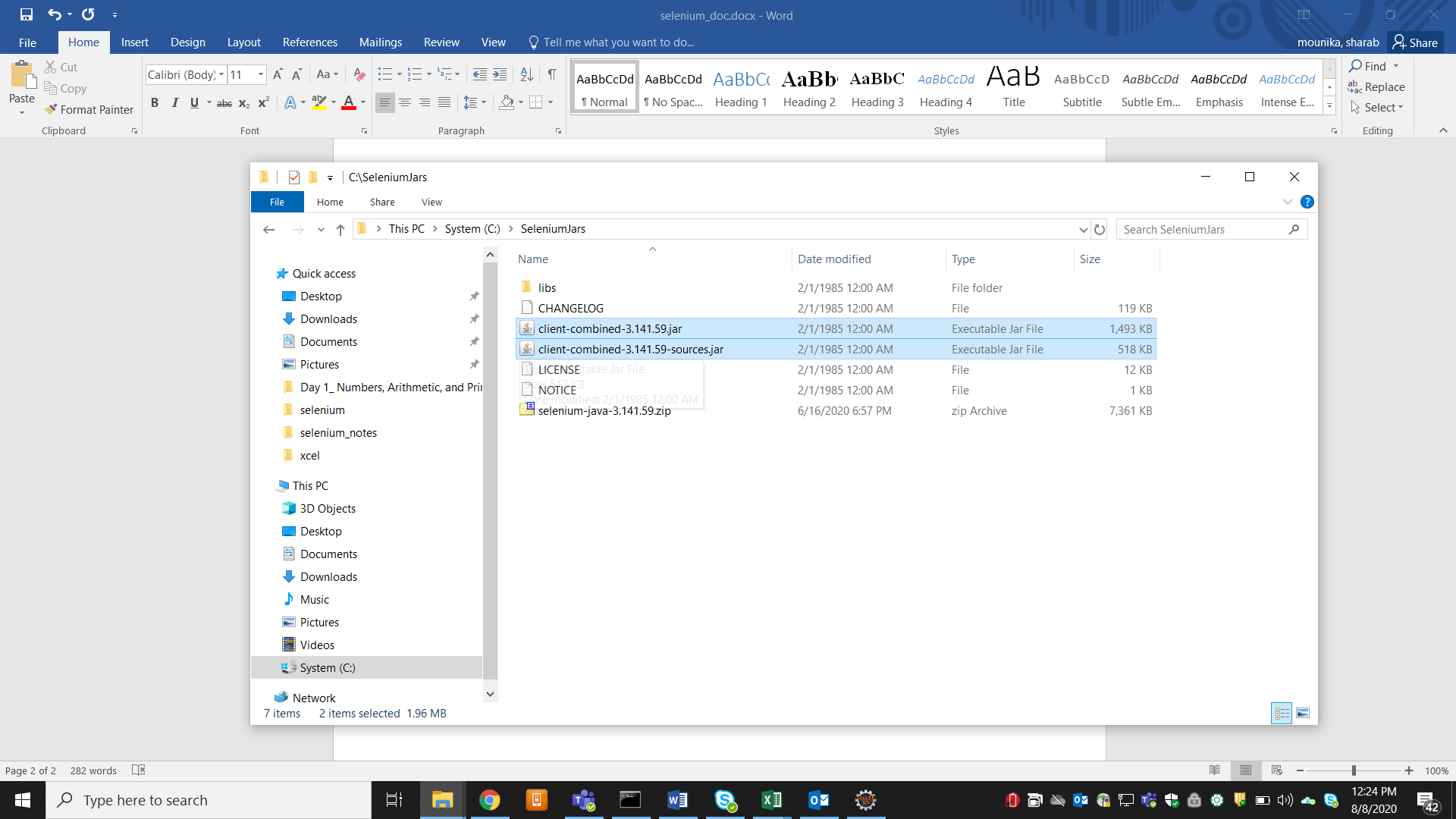
3) download selenium and create a connection between selenium and eclipse

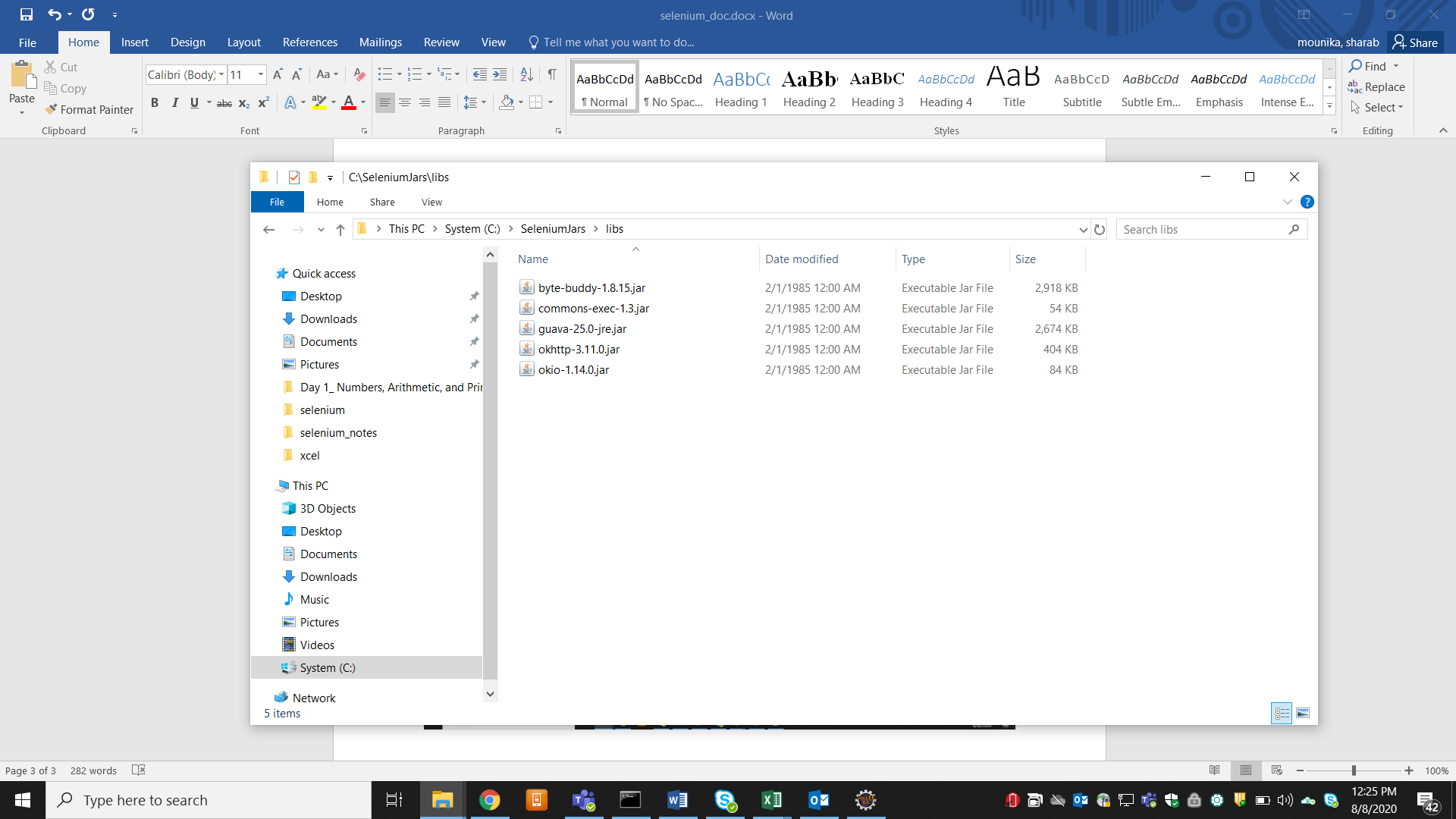
Selenium is jar file which we need to download

<https://www.selenium.dev/downloads/>

download java jar and download alpha(firstone)

right click on proj 🡪properties🡪javabuildpath🡪libraries🡪add external jars under classpath🡪add jars from folder





4) download drivers

Chromedriver is a class which we use if we want to automate chrome browser

Note: web driver is an interface which provides all the methods for your automation we actually taking methods present in web driver so we must strictly implement methods of it so we much give idea to chromedrive object that the return type is webdriver and when you write a class you need to import that specific package and packages will come from jars you have added

Chrome, fixefox etc are third party browers due to some security issues which will restrict selenium to directly invoke them using external api’s so each browser team has given .exe file and asked to invoke that file first

System.*setProperty*("webdriver.chrome.driver", "C:\\DRIVERS\\chrome driver\\chromedriver\_win32\\chromedriver.exe");

Then there after this exe file will inturn invoke browser

WebDriver driver = **new** ChromeDriver(); // chromedriver is a class which is implementing WebDriver interface

// creating object referring to interface to call methods in interface

Selenium Grid:

Selenium-Grid allows you run your tests on different machines against different browsers

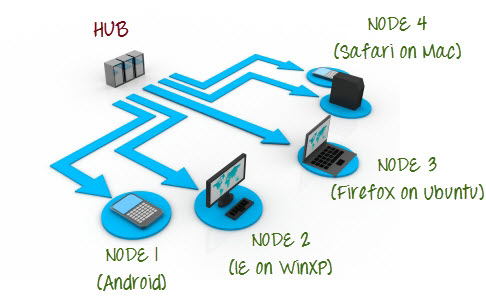
5 testcases

Selenium-Grid support distributed test execution.

You can also run the Tests in Parallel in multiple machines on Selenium Grid

HUB- IE 9 - chrome browser,IE11

50 testcases



**What is Hub?**

* The Hub is the central point that will receive all the test request and distribute them the the right nodes.  
  There should only be one hub in a grid.  
  The machine containing the hub is where the tests will be triggered, but you will see the browser being automated on the node.
* **Node** :  
  Nodes are the Selenium instances that will execute the tests that you loaded on the hub. Nodes can be launched on multiple machines with different platforms and browsers.

Steps

1. Download Selenium server jar-- The Selenium Server is needed in order to run Remote Selenium WebDriver (Grid).

java -jar selenium-server-standalone-3.8.1.jar (use this command in cmd to invoke any jar)

1. Started the hub

java -jar selenium-server-standalone-3.8.1.jar -role hub

1. Login to another machine and register it as node for Hub
2. Download the selenium server jar in node machine as well/ Check java-
3. java -jar selenium-server-standalone-3.8.1.jar -role webdriver -hub >ipaddress>/grid/register -port 5566

eg:IP address is <http://192.168.43.241:5566/> this we will get when we invoke hub command

( Check if java is configured in system variables)

6.Chrome driver.exe file geckodriver-

6.Accessing Hub from Node Machine

http://iporhostnameofHub:4444/grid/console

7.we need to give knowledge to node about drivers

Download chromedriver or geckodriver and place that in the same folder where selenium standalone server is placed

Open cmd

java –D webdriver.chrome.driver=”path of chromedriver” -jar selenium-server-standalone-3.8.1.jar -role webdriver -hub >ipaddress>/grid/register -port 5566

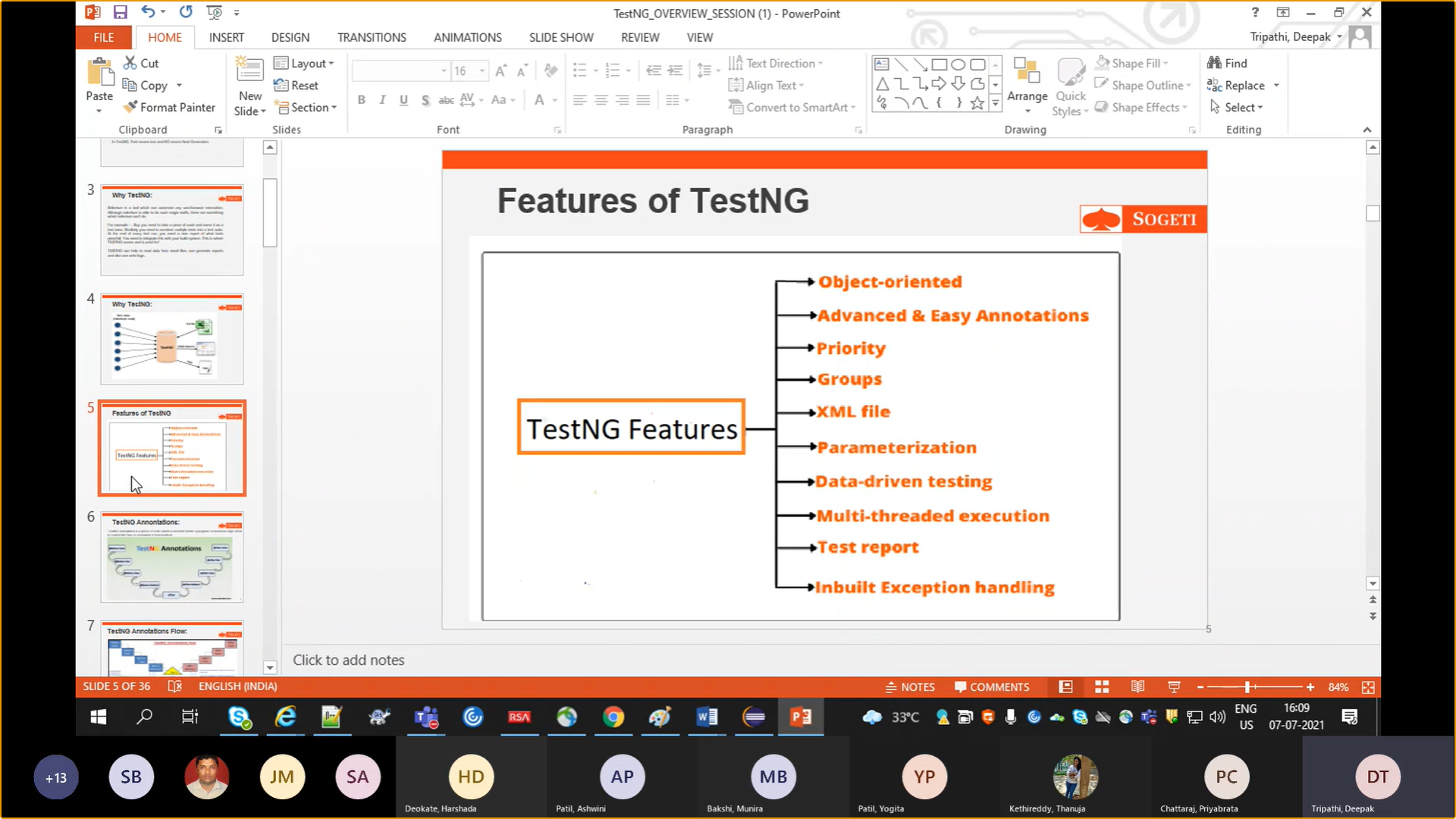
8. now come back to hub(main) and open eclipse and add this standalone jar through exteranal jars through build path

When you want to run on remote machine we will use desired capabilities

TestNG

Advantages of TestNG

* you will have control over execution of tc when I say control it means
  + when one test cases depend on another set of testcases, you can run pre-requisite tc before executing actual test, you can set and map to execute in that way (using helper attributes like dependsonmethod , you can create dependencies , you skip a test using enabled = false helper attribute and give time out to particular testcase)
  + when your manager ask you execute only smoke test, you can group them and execute
  + if you want to exclude or include any tc, you can do that
  + you can do parameterization (you can run one tc with multiple set of data)



Note:

* TestNG itself will act as java complier, you do not need separate java complier to run your test (you need java complier to run main method i.e **public** **static** **void** main(String[] args)) testNG will in turn depends on java on backend
* TestNG needs every tc to be in method and you need to give @Test before every tc, @Test is a annotion in TestNG library which indication which ever method follows it is treated or recognized as tc by TestNG and runs the program even if you don’t provide (public static void main)

<https://testng.org/doc/>

click on eclipse

1-install --🡪install plug-in

Or <https://testng.org/doc/download.html>

* Select*Help / Install New Software...*
* Enter the update site URL in "Work with:" field:
  + Update site for release: <https://dl.bintray.com/testng-team/testng-eclipse-release/>.
* Make sure the check box next to URL is checked and click *Next*.
* Eclipse will then guide you through the process.

1. In single class you can create no of testcases we need to mention it as @test before it
2. To create XML file Right click on project--🡪testng---🡪converttoTEstNG
3. Suit----🡪Test(module)----🡪 classes (no of testcases)--🡪methods(testcase)
4. Suit can have multiple module and each module can have no of class and each class can have no of testcases inside it
5. Right click on xml file and run as tesng to run all tests
   * You can define no of testcases in a single class
   * You can modularize based upon functionality and trigger them accordingly (creating test modules as per requirement)
6. To exclude any method in the class use

<methods>

<exclude name=*"mobilecarloan"*/>

</methods>

7.if there are 20 tc but if you want to execute only 1tc in that class then use include

<methods>

<include name=*"test1"*></include>

</methods>

By using include or exclude we will have control over running specific tc in a class

Based on requirement we can quickly create an xml file and execute

8) if there are 100 tc and you want execute only 70 and exclude 30 in that case we will use regular expressions. We will be having naming conventions for tc names we need to use that names to write regular experession

<methods>

<exclude name=*"mobile.\*"*></exclude>

</methods>

Tc starting with mobile name will be excluded

9) we can execute at package level also if there are 10 package if you want to execute only one package

<suite name=*"Loan Department"*>

<test name=*"Personal loan"*>

<packages>

<package name= *"TestNGPackage"*/>

</packages>

</test>

</suite>

10)Annotations

* BeforeTest

Test here refers to the test(module) in xml file that means this is the first tc need to be executed in the test folder in xml file and scope will be till the test folder or module

Eg: cleaning up the database before start executing

* AfterTest

Similar to beforetest, aftertest will be executed after all tc’s in the particular Testfolder(module)

Note : if there are two or 3 testfolders(modules) this beforetest and aftertest will be executed for before and after testfolder(module) where beforetest Tc belongs to but not for all modules

* @BeforeSuite: method or Tc under beforesuite will execut first before any testcase in XML file, control is over entire xml file
* @AfterSuite: method or Tc under beforesuite will execut at last after execution of all testcases in XML file
* @BeforeMethod : it is confined to class file if there are 5 methods(tc) in the class, method under @beforemethod will be executed before all 5 methods
* @AfterMethod: it is confined to class file if there are 5 methods(tc) in the class, method under @Aftermethod will be executed after all 5 methods

Note : Beforetest is related to xml file, scope is restricted to test module in xml file and beforemethod is related to classfile and scope is restricted to the class in which before method is written

* @BeforeClass: it is confined to class level. Method under beforeclass will be executed before any test or method in that class
* @afterClass: it is confined to class level. Method under beforeclass will be executed after all execution completion of all tests or methods in that class

Note: beforemethod and beforeclass both confined to class level and @test will be executed as alphabetical order

11)Groups: if there are 100 tc in different classess and you want execute only 5 tc out of 100 here we will use the concept of groups

Eg : out of 100 tc out of which 5 are considered as smoke test, which needs to be run everyday or after receiving new build

Xml eg:

<suite name=*"Loan Department"*>

<test name=*"Regression"*>

<groups>

<run>

<include name=*"smoke"*></include>

</run>

</groups>

What we have mention in @test

@Test(groups= {"smoke"})

**public** **void** mobilehomeloan() {

System.***out***.println("mobilehomeloan");

}

In similar way we can exclude only smoke tc’s use exclude instead of include in groups

11)Annotation helper attribute

\*)Actually @Test will be executed alphabetically but if you want execute particular test say TestA before TestB has TestB execution depands on TestA in that case we use helper dependsOnMethods

@Test(dependsOnMethods= {"test1"}) // if there are multiple tc dependency, give a comma and give names of menthod or tests in {}

**public** **void** test2() {

System.***out***.println("personal loan second test case");

}

\*)Enabled: if there is a bug or errors in test or flow and everyone knows about it so don’t want to run that test or skip it you can do it using enabled

@Test(enabled=**false**)

\*) timeOut: if you think there is one particular test which takes more time

@Test(timeOut=4000)

**public** **void** Webcarlon()

{

System.***out***.println("Webcarlon");

}

12) Parameterizing from TestNG xmlWel file

If you think there is global variable which is constant you can create that parameter in testing xml file

In xml file

<suite name=*"Loan Department"*>

<parameter name=*"URL"* value=*"http://google.com"*/> //this is at global level

How you use that variable in test

@Parameters({"URL"})

@Test

**public** **void** a(String url) {

System.***out***.println("personal loan a test case");

System.***out***.println(url);

}

Only under that test this parameter is accessible

Note: if you define the same url or can override under particular test(module) in xml file which can be accessible to all classes under that test(module)

<test name=*"homeloan"*>

<parameter name=*"URL"* value=*"https://rahulshettyacademy.com/angularpractice/"*/>

You can pass multiple parameters at testfolder level as well

<test name=*"homeloan"*>

<parameter name=*"URL"* value=*"https://rahulshettyacademy.com/angularpractice/"*/>

<parameter name=*"APikey"* value=*"12345"*/>

@Parameters({"URL","APikey"})

@Test

**public** **void** WebHomelon(String url,String api)

{

System.***out***.println("Webhomelon");

System.***out***.println(url);

System.***out***.println(api);

}

13)dataprovider annotation(parameterizing tetscases)

If there are parameters which have to be sent at testcase level

Parameterizing with multiple datasets by running tests with multiple combination

We are not doing anything xml file directly we are creating a method under annotation @DataProvider then we are using this method or passing this method to test which has be run which different number of combination of data

@DataProvider

**public** Object[][] getdata() {

//creating multidimentional object array

//Object[][] data =new Object[3][2]; //row – how many data sets or how many times it should run and column – how many values in each set

Object[][] data = {{"mounika","25"},{"a","23"},{"b","20"}};

// as you are sending 3 sets of data each set having two parameters username and age then corresponding test will run for 3 times with 3 different sets of data

**return** data;

// following test will run for 3 different data sets

@Test(dataProvider="getdata")

**public** **void** datademo(String username,String age) {

System.***out***.println("data provider annotation");

System.***out***.println(username+" "+age);

}

Note: how do you achieve parameterization? We can do it through xml file(giving commonly used parameters at global or test level) or directing in test using @dataprovider annotation

14)TestNg listeners

Listeners will listens to your test execution whether your tc is getting passed or failed and if in case your testcase has failed it will invoke one method to take SS on failure

//ITestListener is a interface which has methods of this TestNG listeners so you need to implement that interface to use listeners methods

Eg: there is method say onTestFailure so we can write some screenshots code in this method so whenever our test fails testing will reroute to this listerner method to execute that method

We need to tell testing xml file where exactly listerners class is located if not it will not redirect

<suite name=*"Loan Department"*>

<listeners>

<listener class-name=*"TestNGPackage.Listeners"*/>

</listeners>

And if you want to get name of test which got failed use ItestResult class which has info of result using getName() method we get the name of test

@Override

**public** **void** onTestFailure(ITestResult result) {

// **TODO** Auto-generated method stub

//screenshot code

//response if API is failed

System.***out***.println("I failed executed Listeners Pass code" +result.getName());

}

15) Running tests parallel with testing

Give parallel=”test”

<suite name=*"Loan Department"* parallel=*"tests"* thread-count = *"5"*> this is at test folder level or suite level

Thread-count is no of test needs to run in parallel

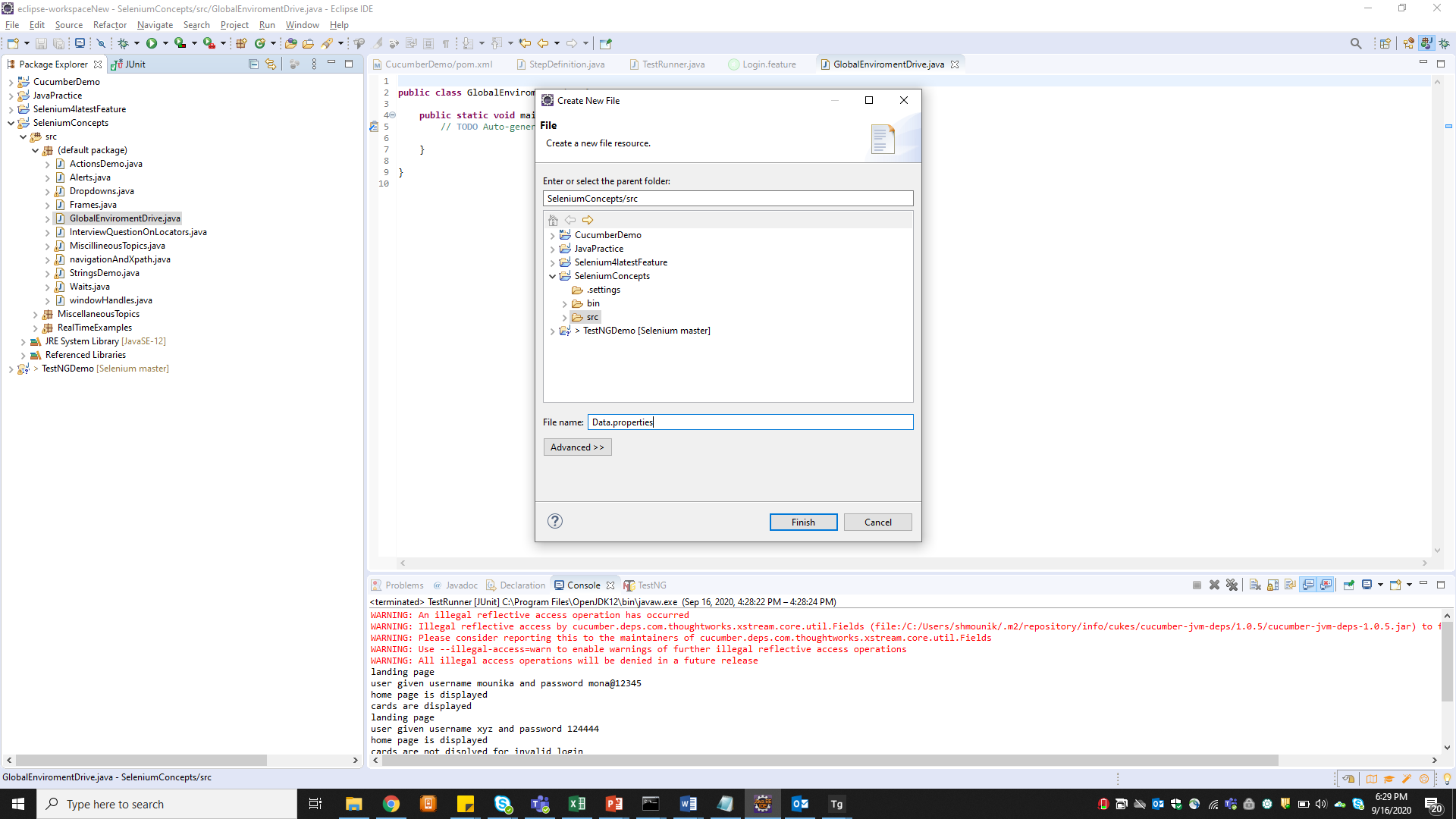
You can run classes at parallel by giving it in test folder

<test name=*"Personal loan"* parallel=*"classes"* thread-count = *"2"*>

Note: you can see testing report go to output folder there will be index.html open that in chrome

Framework2-Global Environmental

variables Setup and Reusable components

1. To set variables globally eg: for browser,timeouts,urls etc.
2. Create a file with properties extension 

Framework3-Maven

Maven –

Apache Maven is a software project management and build management tool for Java Frameworks.

1. Why Maven?

* Central repository to get dependencies

That means if there are 50 people in your project and if you want to share your project with them even if you share jars which you have added locally will not be available for them in real time if we use maven project where you add dependencies (pom.xml file) this will go to maven repository and download required jars for us

* Maintaining common structure across the organization

That means maven suggests structures or templates or skeletons for eg. For java project it will suggest on template so for maintaining consistency most of projects uses maven

Maven gives skeleton for cucumber framework

Maven not only gives cucumber skeleton but also many other skeletons for web development as well, you just need to tell which kind of structure you need, it will provide you that kind of skeleton

You can create your own structure but it will take a lot of effort in creating packages , classes folders etc, but when you tell maven it will give you readymade skeleton

* Flexibility in Integrating with CI tools

Let’s say there are 500TC in your project and you want run overnight and see the results next day morning and for that you need build management tool to integrate with CI tool (Jenkins) to schedule at particular time to execute all tcs’s

* Plugins for Test framework execution.

1. Install Maven

* First you need java and set system variables for java
* Download (binary zip folder)

1. Set System variables to recognize Maven

* Set Maven home(C:\Users\shmounik\apache-maven-3.6.3) and Bin path in path variable (C:\Users\shmounik\apache-maven-3.6.3\bin) in system variable
* Type (mvn –version) in cmd

1. Understanding Maven terminologies

**Artifact**: An artifact is a file, usually a JAR, that gets deployed to a Maven repository.

**GroupId**: groupId will identify your project uniquely across all projects,

// groupID in dependencies is unique ID given to each jar and artifactID is sub project ID for eg if you want jar for selenium (groupID) and that too for java (selenium java ) that is artifact sub project

**archetype:generate** ; Generates a new project from an archetype

1. Creating Maven Project in eclipse for automation project select 🡪 “quick start” and artifact Id will be project name you need to groupID as well package name will be groupID.artifactId if you want you can change the package name
2. Understanding POM.xml file dependencies..

* Search for selenium java and testNG dependencies in maven repository and add them in pom.xml file

1. Importance of Maven Surefire plugin: to execute all your tests in maven project you need this plugin. By default you will not get that in skeleton to need to go to their official website. Search in google surefire plugin-🡪usage-🡪

<build>

<pluginManagement>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>3.0.0-M5</version>

</plugin>

</plugins>

</pluginManagement>

</build>

1. Add this in plugin in pom.xml above dependency
2. Maven Phases for Test Automation

You have be in that project folder in cmd and follow below command to execute test

1. Clean, compile, test

* mvn clean : cleans all builds(it deletes all temporary files in the current project)
* mvn compile: it will compile that means it scans and checks for syntax errors
* mvn test : Actually triggers the test execution ( even if you don’t give clean and compile and just give test maven will do clean and compile in background)
* and when you hit Test command first it will read for dependencies in pom.xml file and goes to .m2 repository in your local and checks all the dependency jars are brought by maven earlier ( whenever it sees dependencies it will not directly goes to maven repository first it will check in local repository which is automatically created when you setup maven
* one important point if you want maven to recognize you test then you need to give class name ending with Test word i.e seleniumTest.java as maven uses algorithm which expects Keyword “Test” at the end of each java class file ( if you don’t with “Test” word then maven will not execute that java class test

1. Integrating Testng into Maven

* mvn test : will execute complete test folder by default
* now create xml file for testNG and run that xml file through maven commands
* right click on project🡪testNG🡪 converttotestNG🡪testngxml file will be created

1. Testng xml files configuration in POM file

* You need to change add configuration setting from Using Suite XML Files add that in surefire plugin before version (Another alternative is to use TestNG suite XML files. This allows flexible configuration of the tests to be run. These files are created in the normal way, and then added to the Surefire Plugin configuration:))
* Now when you run mvn test command it will run testng xml file
* Now when you run mvn test first it will come to pom.xml and read the instructions then it triggers
* To run single test from maven use command “mvn -Dtest=TestCircle test”

1. Setting up Profiles in POM.xml file

* Suppose there are two Testngxml files one for regression and one for smoke
* If above is the case we will build profiles here
* Create parent profile the create two child profiles with two ids and paste the build code(which is having which testng xml file need to be executed ) in both profiles and change testng xml file names in the builds

1. Running selected Tests only with Maven commands

* Mvn test –PRegression
* -P means profile and you need to give ID of profile which you want to execute

1. Maven Sure fire reports

Taking screen using Ashot API

Add Ashot dependency in maven pom.xmlfile (search in google ashot dependency )

<!-- https://mvnrepository.com/artifact/ru.yandex.qatools.ashot/ashot -->

<dependency>

<groupId>ru.yandex.qatools.ashot</groupId>

<artifactId>ashot</artifactId>

<version>1.5.4</version>

</dependency>

Go to test add below

1)For complete page

Screenshot screenshot = new AShot().shootingStrategy(ShootingStrategies.viewportPasting(1000)).takeScreenshot(driver);

ImageIO.write(screenshot.getImage(), "jpg", new File(".\\screenshot\\ashotimgelement.jpg"));

2) for particular element( pass element)

WebElement ele =driver.findElement(By.xpath("/html/body/div[4]/div/header/div[2]/div/div[1]"));

Screenshot screenshot = new AShot().shootingStrategy(ShootingStrategies.viewportPasting(1000)).takeScreenshot(driver,ele);

ImageIO.write(screenshot.getImage(), "jpg", new File(".\\screenshot\\ashotimgelement.jpg"));

FrameWork 4 Jenkins:

Jenkins is continuous integration tool for automation ( automation doesn’t means automating web UI basically daily routine things like building jars ,pushing code into the website , running all automated testcases it can be anything without manual effort )

* Real time example where you will use Jenkins : if your all your testcases execution takes time of 7hours and then we will create a job in Jenkins and schedule that at midnyt at 2am and by the time you reach office by 9 am your test execution will be completed
* If there are two people running the code in their individual machine there is a chance both might get different results some may use latest testng version other may use selenium latest version so to maintain consistency and remove all dependencies when you trigger tests from Jenkins, Jenkins will have its own environment and if you pass stable environment there it will automatically run on setting provided there

Jenkins installation

* Basically we don’t have server to host usually company will provide us
* Download Jenkins ->generic java package(.war) as we are Using Jenkins as windows service not hosting on any servers
* Go to location where jar is located and do cmd from there and type command “ java –jar jenkins.war –httpPort=8080” // use 9090 if 8080 is busy

Configuration global settings in Jenkins

* Click on manage Jenkins-🡪 global tool configuration
* Click on jdk installation(for the first time) and give the path of java\_home
* Similarly click on maven and give path of maven\_home

Understanding Jenkins workspace

* How to place you project in Jenkins : first you need to place your project in jenkins home directory (copy your maven project (once you run jenkins you will see “ .jenkins “ open that folder and paste your project there bcoz that is home directory of jenkins so when you run you job jenkins comes and checks for your project in this directory
* Create new job-🡪 click on new item 🡪give name (mavenJOb) and select freestyle project -->it will be navigated to page where you need tell from where to want to get the code
* Usually we will be getting it from git(server link) and run but now we will get from maven from local machine
* So click on advance settings🡪 use custom workspace -> ${JENKINS\_HOME}/PROJECTNAME
* ${JENKINS\_HOME} will give you path till jenkins home directory
* Now go to “Build”-🡪 select “ invoke to level maven targets” -🡪 select maven version(if you have configured maven properly)-🡪given maven command in your goals tab(test PRegression) no need of giving mvn in command

To execute job from Jenkins

* Click on “build now” once build is completed click on “ click on console output

TestNG Reports for Jenkins Jobs

* Actually you can see testng reports (target 🡪surefire-suite----🡪index.html
* You can also see testng.result.xml file if you want this xml reports in jenkins as well you need to download a plugin in jenkins
* TestNg cameup with one plugin TestNG results plugin if you install it jenkins will generate reports
* To install that plugin go to jenkins🡪 manage jenkins -🡪manage plugins-🡪Search for testng results plugin -🡪 install
* Now again if you want to change setting for your job click on “ configure”
* Go to “post build actions” 🡪publish testNG results and save it
* Again build now once after you added plugin now you will see “TestNG results” click on that -🡪 click on “(build#) “ then you can see results
* If run same jobs everyday it will show graphs also

Test Job Scheduling

* Go to configure
* Now go to “Build Triggers” ( you can schedule your jobs here)-🡪 build periodically🡪(basically there is format to give it like you need to give in( Minutes Hours(0-23) DayOfMonth(1-31) Month(1-12) DayWeek(0-7))
* You can just give hour like (\* 7 \*\*\*) that means it will run at 7am every day

Extent Reports

* Extent reports are used to receive execution results
* Create a sample maven project
* Then search for extent report maven dependency and add that dependency in pom.xml

<dependency>

<groupId>com.aventstack</groupId>

<artifactId>extentreports</artifactId>

<version>5.0.4</version>

* </dependency>And also add selenium-java and selenium-testng dependencies as we

FrameWork-5 Excel Data Driven Testing utilities

* Apachi Poi is a api to connect with excel and pull the data from excel
* You can download Apache POI from below official link <https://poi.apache.org/> but we are not downloading any jars we will directly create maven project(click on simple project(skip archetype selection))and add apache poi and poi ooxml dependencies

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>4.1.2</version>

</dependency>

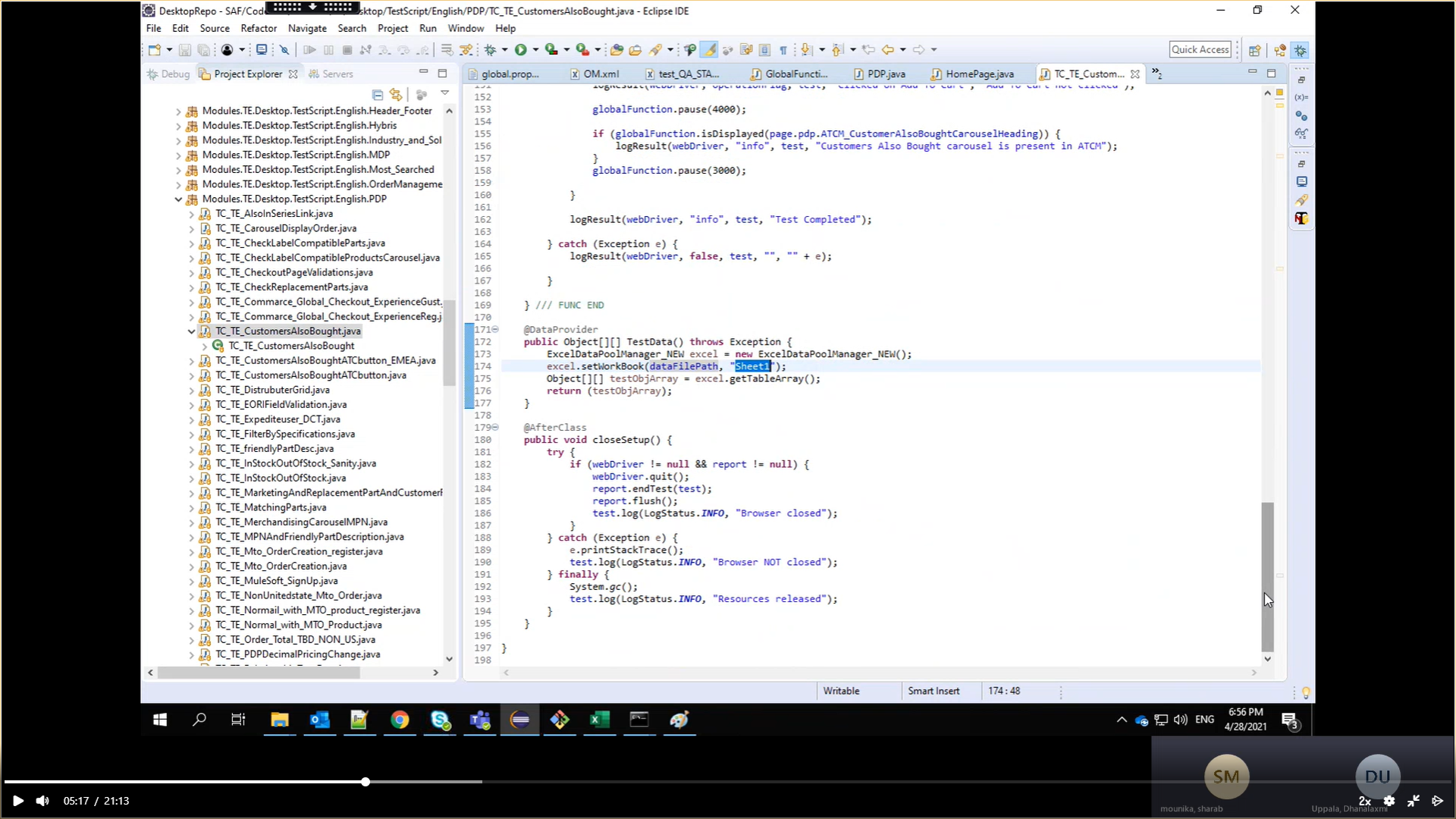
<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>4.1.2</version>

</dependency>



FrameWork6-LoggingFramework-log4j

Log4j is a reliable, fast and flexible logging framework(API’s) written in java which is distributed under apache software license (it is a utility given by apache guys which can be used in any testing frameworks eg. Selenium, appium, REST API etc. which uses java as programming language, even web development teams uses this for logging purpose)

Why log4j?

It can be used to answer following questions

Log only when there is error in package A---

<Loggers>

<Logger name=*"packageB.classB1"* level=*"error"* additivity =*"false"*>

<AppenderRef ref=*"Console"*/>

</Logger>

<Root level=*"trace"*>

<AppenderRef ref=*"Console"*/>

</Root>

</Loggers>

// if you don’t give addivity=”false” it will run two times once with the logger and one more time with normal root level and when you are giving addivity=”false” in logger it will skip executing in root level

* Log entire message for package B – use trace for level in root
* I want to log with time stamps -- patternlayout
* I want the log file of last week
* How will I know there is error by just looking at logs

Importing log4j into our project

* Search for “download apache log4j jar file”
* Download binary zip file
* Create a java project
* We need to add only two jars( 1)Log4j-api-2.8 2)log4j-core-2.8)

Need to define Configuration file

* Go to this link <https://logging.apache.org/log4j/2.x/manual/configuration.html> where they explained how to create log file
* Create a XML file
* Depending on configuring file only log will act (there are two main tags – Appenders, loggers)
* We need to ask 3 questions
* where to log – Appenders Tag is used \_console
* what to log – this will happen in loggers Tag(logger and root tag level )
  + root tag – if you give level as error it will print only errors and if you give level as trace it will print all logs (root level means for all packages)
  + loggers- in loggers we can target specific packages and give level = trace or error
* how to log – patternlayout
* create a separate folder say resource(not under src) and create a separate xml file(copy the default code there) under it, now add this folder to source folder through build path --- buildpath🡪sourcefoder🡪add folder ( when you run a test, it will go and check in build path and comes to know about configuration file)
* if you want to log all your result in one folder then create a folder( say logs) and create a file (say print.log)and give that file path in appender tag refer eclipse code , basepath means – path will till project level , you have given till .log path in filename

first 500kb file will be stored in print.log file after in file pattern we will pattern how next file name have to be created eg : <RollingFile name=*"File"* fileName=*"${basePath}/prints.log"* filePattern=*"${basePath}/prints-%d{yyyy-MM-dd}.log"*>

<PatternLayout pattern=*"%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"*/>

<SizeBasedTriggeringPolicy size=*"500"* />

* </RollingFile>
* triggeringpolicy size 500 means once file reaches 500kb it will create new file for reports

When to use log4j methods based on given scenarios

\*\*\*\*\*\*\*When to Use log. Error, debug and info methods in Selenium test cases\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Use log. Error() to log when elements are not displayed in the page or if any validations fail

Use Log. Debug()

When each Selenium action is performed like click, SendKeys, getText()

Use log.info()

When operation is successfully completed ex: After loading page, or after any successful validations

It’s just counterpart to log. Error()

FrameWork7-PageObjectPattern&pagefactory

Page object pattern

* Easy to maintain (means when you are using one object in 100 scripts, if that single object is changed and it should be such that if I change in once place it should reflect in all the places not like going to 100 scripts and updating them)
* Easy readability of scripts
* Reduce or eliminate duplicity
* Re-usability of code

Eg: if there is website so we will be creating object class for every page or functionality eg : we will be writing objects of complete home page in one class and when we are creating testcases if you want to use any object we will be calling that object in our test

Advantage is if there is change in object of homepage if we change in one place is enough instead of 100 testcases using that object.

1)declaring -> By username

2)initializing ->

3)Utilizing -

Creating Page object constructor in class

* Create a java project
* Create two package one for page object repository and one for testcases
* See selenium code for example

Refer selenium for page factory code

Framework 8 - CuCumber

TestCases Specifications in Behaviour Driven Development :

Step 1:

Testcases should be defined in a Business level by following a ubiquitous language

Ubiquitous language is a (semi-)formal language that is shared by all members of a software development team — both software developers and non-technical personnel.[15]

Syntax :

In order to(Achieve something)-> Business outcome

As a (Role)

I want to Do this

Example:

In order to Pay Credit Card Payment

As a NetBanking sole owner who have credit section access

I want to Navigate to credit card section, Enter amount and process my Payment

Step 2:

Its time to Decide what Test cases are sufficient to certify Business Scenario working

Syntax for Writing Testcases:

Given (what you need to have to perform action), -Prerequists

When (performs action)- Action

Then (the desired outcome for the user).- Validation

Example:

Given : An account with zero balance

When: I navigate to Credit card Payment Section and click on submit bu giving amount

Then : It should throw error message –fund

Given : An account with sufficent balance who does not have credit card

When: I navigate to Credit card Payment Section

Then :You don't have to access as your are error message

Given : An account with sufficent balance who does not have credit card

When: I navigate to Credit card Payment Section and amount

Then :You don't have to access /error message

Advantages:

This can be used as Standard Template where all QA can stick to one common standards of defining Testcases (everyone has his own style of writing testcases, it may difficult to understand sometimes, if everyone uses same format of writing tc , it will be easier to understand)

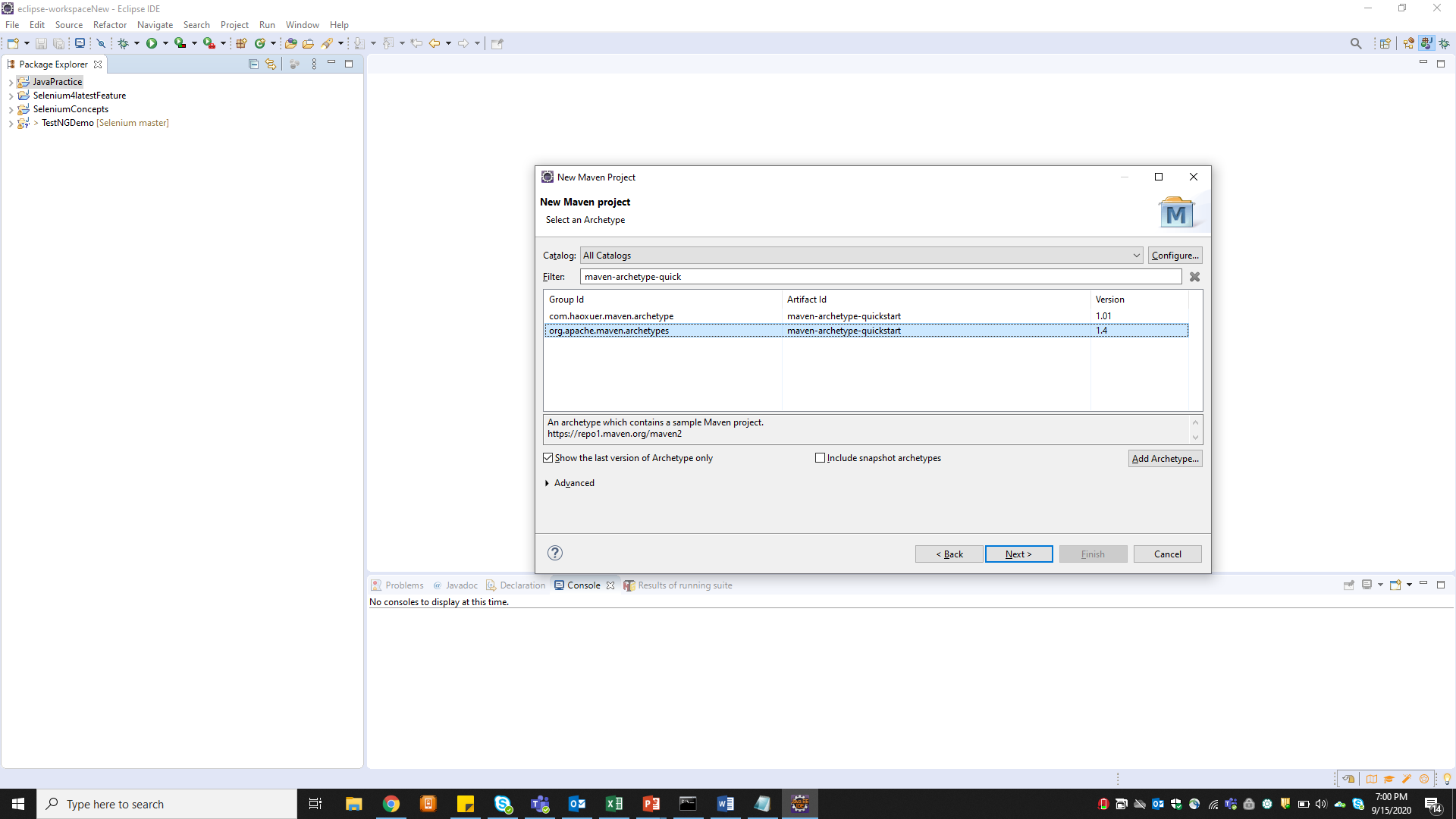
Each Scenario reflects a Business Value

We can estimate the Test coverage happened for Each Business outcome by going through Test Cases

We can tag these Annotations to Selenium Automation and execute the Business Testcases

Common Standardised Testcase template for both Manual and Automation testing

\*) Creating cucumber project

1. We need to install a plugin to support gherkin language
2. To download plugin=Help->marketplace-> type natural and download that natural plugin
3. We need to get cucumber project skeleton from maven (you cannot simply create simple java project,cucumber expects a proper structure for its project)
4. Create new project->maven project🡪 select “quick start template” as cucumber supports that
5. 
6. Give groupID(project name) and artifactID(package name) now a cucumber project structure will be created

\*) Cucumber Dependencies

1. When you have created Maven project (for cucumber skeleton) POM.xml will be created. Basically there will be server called maven repository backend where all software jars are located so whichever jars related to that project must be placed in that POM.xml file now project will talk to the maven repository through this xml file, now in this case we need to add cucumber jars in this pom.xml file
2. For cucumber we need two jars (1.Cucumber-java 2.cucmber-junit)
3. We need to give dependencies info in xml file (no need of downloading jars and uploading)
4. Search cucumber-java maven repositories it will show you

<!-- https://mvnrepository.com/artifact/info.cukes/cucumber-java -->

<dependency>

<groupId>info.cukes</groupId>

<artifactId>cucumber-java</artifactId>

<version>1.2.5</version>

</dependency>

5)now search cucumber-junit

<!-- https://mvnrepository.com/artifact/info.cukes/cucumber-junit -->

<dependency>

<groupId>info.cukes</groupId>

<artifactId>cucumber-junit</artifactId>

<version>1.2.5</version>

<scope>test</scope>

</dependency>

\*) Cucumber framework terminologies

1. Feature file 🡪where you define your testcases which need to be automated high level or plain english language
2. Step definition🡪 automation code for the feature file
3. Junit TestRunner🡪to connect and trigger the tc we will use this
4. Now create a package(for feature) under src/test and under that package create a file and always end file with .feature
5. Now create one more package for (step definition) and create a java class under that package
6. Now map feature file with step definition class file
7. 🡪 suppose “*Given* User is on Netbanking landing page”is in feature file
8. Now in set definition you have to give like @Given("^User is on Netbanking landing page$") start with ^ and end with $
9. You can actually create set definition file when you give feature file in tidy gherkin plugin
10. Search tidy gherkin in google Go to tidy gherkin launch app and paste feature file and click on java steps then you will get basic step definition steps without code
11. For testrunner again create one more package and create a java class under that
12. There is one more way to get basic stepdefinition file for respective feature you just map feature file and Stepdefinition file without any code and run testrunner then on console you will get basic stepdefinition implementation

6)Add page object and handle dynamic popups in homepage

End to End real time project framework

* Create maven project and configure framework dependencies
  + Search “selenium maven dependencies
  + Add selenium-java, TestNG
  + We will be creating all testcases under src/test/java and page objects will go into src/main/java
* Create global variable setup
* Create page object repository for all objects
* Use testNG annotation before methods or test
* Integrate testng xml file in maven pom.xml (go to maven official website Using Suite XML Files – add plugins between <build>
* Now execute through maven commands
* Integrate log4j into framework (add log4j maven dependencies instead off adding external log4j jars) and here no need of creating separate folder for log when you run maven will create a logs folder (if you are not getting any result try to add resource (where log4j xml is present )folder to source folder through buildpath)
* Add log4j xml file in resource folder
* We have to tell our maven pom.xml about this log4j xml file by adding resource filtering plugin in build in pom.xml

<resources>

<resource>

<directory>src/main/resources</directory>

<filtering>true</filtering>

</resource>

</resources>

* Using testNG listeners to take screenshots on failure of test (create a java class say listeners)

public class ListernersClass implements ITestListener{

}

Even though you have implemented interface if you don’t see any overridden methods

Rightclick on class-🡪 source🡪 override/implement methods

And give the path of listener’s class to testng xml file

* Next is Extent reports -🡪 add extent report maven dependency and create separate java class all extentreports
* In listeners we have onteststart method(which will be executed before executing the actual test) we will pass extent.createTest to this method
* Integrate maven with Jenkins
  + How to parameterize jenkin build with different browser execution?
  + We have to set browser property from maven commands(mvn test –Dbrowser=chrome
  + Change one step in code “String browserName =System.getProperty("browser");”
  + Now go to project in Jenkins –configure -🡪”This project is parameterized”-🡪add parameter🡪choice parameter-🡪(given name as browser , choices as – chrome,firefox,safari,IE ,chromeheadless ect) -🡪save
  + Once you did that you can see option “build with parameter “ option above configure in your project when you click on that you can see a dropdown showing your choices but we need given that knowledge maven command to each option
  + Now to configure again and earlier you have given maven command right now change it to “mvn test –Dbrowser=”$browser” // here $browser is Jenkins variables you have created in parameter
  + Now again go to build parameter choose any browser and now your test will run in that browser
* Coding standards to achieve encapsulation in frame work
  + Defining private locators and accessing them through public methods (this is encapsulation)

Remove hard coded paths in framework by generating paths dynamically

* + System.getproperty(“user.dir”) will give you path till project
  + Eg: FileInputStream fis=**new** FileInputStream fis=**new** FileInputStream(System.*getProperty*("user.dir")+"\\src\\main\\java\\resources\\data.properties");
* How to Execute test in chrome head less mode
  + See code in browserInvocation java class
* Where do we use inheritance,interface and statics are used in framework?
  + Inheritance – we are inheriting browserinvocation class in every other test class
  + Interference – for testNG listerners we are implementing Itestlistener interface
  + Statics – static basically means if you created a variable as static if multiple tests are accessing the variable all objects share common property basically new copy will not be created once one test has used it and again whether another test is using it it will override the variable
  + Main advantage is no memory is wasted
  + If you are executing in sequential manner static can be advantage but if you are executing in parallel, then it will throw null pointer exception as variable will be overridden and we can handle this see code in homepage (here we have created local variable as soon as test starts static variable will be assigned to local variable)
* Giving our test cucumber knowledge –
  + Now create three packages for features, step definition and test runner
  + Add cucumber dependencies
  + Parameterizing cucumber means running with different data
    - When you are using parameterizing change scenario to scenario outline see code in feature file
    - Once that is done we need to change step definition also no need of remembering syntax just run test runner it will give you syntax for step definition copy there or use tidy gherkin plugin and paste feature file code
  + Integrate cucumber with testng and maven
    - Give testrunner path in testng xml file
    - Add cucumber-testng maven dependency
    - Make sure cucumber java and cucumber testng versions are same
    - Now go to testrunner class and extend AbstractTestNGCucumberTests
    - update project (right click –maven—update
  + to integrate cucumber to maven add “ <suiteXmlFile>testngCucumber.xml</suiteXmlFile>” in configuration of pom.xml file
  + now run run through maven commands

DataBase connection to selenium testcases

* We need mysql editor and server ( search for mysql installer and download it will come with both editor and server download community one not web community
* ghgg