



JSPM's
Imperial College of Engineering and
Research (ICOER)

LABORATORY MANUAL

AY: 2023-24

Laboratory Practice IV
410245(D) Software Testing and Quality Assurance

BE Computer Engineering
Semester -I
Subject Code – 410247

TEACHING SCHEME	CREDIT	EXAMINATION
Practical: 2Hrs / Week	01	TW: 50 Marks

Contents

Sr. No.	Problem Statement	Page No.
410245(D) Software Testing and Quality Assurance		
1	Write TEST Scenario for Gmail Login Page	3
2	Test Scenario for Gmail – Inbox Functionality	8
3	Write Test cases in excel sheet for Social Media application or website	10
4	Create Defect Report for Any application or web application	17
5	Installation of Selenium grid and selenium Web driver java eclipse (Automation tools).	28
6	Mini-Project 1: Create a small application by selecting relevant system environment/platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Prepare Test Cases inclusive of Test Procedures for identified Test Scenarios. Perform selective Black-box and White-box testing covering Unit and Integration test by using suitable Testing tools. Prepare Test Reports based on Test Pass/Fail Criteria and judge the acceptance of application developed.	36
7	Create a small web-based application by selecting relevant system environment/platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Narrate scripts in order to perform regression tests. Identify the bugs using Selenium WebDriver and IDE and generate test reports encompassing exploratory testing.	77

EXPERIMENT No: 1

Title: Write TEST Scenario for Gmail Login Page.

Test Cases – Login Page

Following is the possible list of functional and non-functional test cases for a login page:

Sr. No.	Functional Test Cases	Type –Negative/ Positive Test Case
1	Verify if a user will be able to login with a valid username and valid password.	Positive
2	Verify if a user cannot login with a valid username and an invalid password	Negative
3	Verify the login page for both, when the field is blank and Submit button is clicked	Negative
4	Verify the “Forgot Password” functionality	Positive
5	Verify the messages for invalid login	Positive
6	Verify the “Remember Me’ Functionality	Positive
7	Verify if the data in password field is either visible as asterisk or bullet signs.	Positive
8	Verify if a user is able to login with a new password only after he/she has changed the password.	Positive
9	Verify if the login page allows to log in simultaneously with different credentials in a different browser	Positive
10	Verify if the Enter key of the keyboard is working correctly on the login page	Positive
Other Test Cases		
11	Verify the time taken to log in with a valid username and password	Performance & Positive Testing
12	Verify if the font, text color, and color coding of the Login page is as per the standard.	UI Testing & Positive Testing
13	Verify if there is a ‘Cancel’ button available to erase the entered test.	Usability Testing
14	Verify the login page and all its controls in different browsers	Browser Compatibility & Positive Testing.

Non – functional Security Test Cases:

Sr. No.	Security Test Cases	Type – Negative / Positive Test Case
1	Verify if a user cannot enter the characters more than the specified range in each field (Username and Password).	Negative
2	Verify if a user cannot enter the characters more than the specified range in each field (Username and Password).	Positive
3	Verify the login page by pressing ‘Back button’ of the browser. It should not allow you to enter into the system once you log out.	Negative
4	Verify the timeout functionality of the login session	Positive
5	Verify if a user should not be allowed to log in with different credentials from the same browser at the same time.	Negative
6	Verify if a user should be able to login with the same credentials in different browsers at the same time.	Positive
7	Verify the Login page against SQL injection attack	Negative
8	Verify the implementation of SSL certificate	Positive

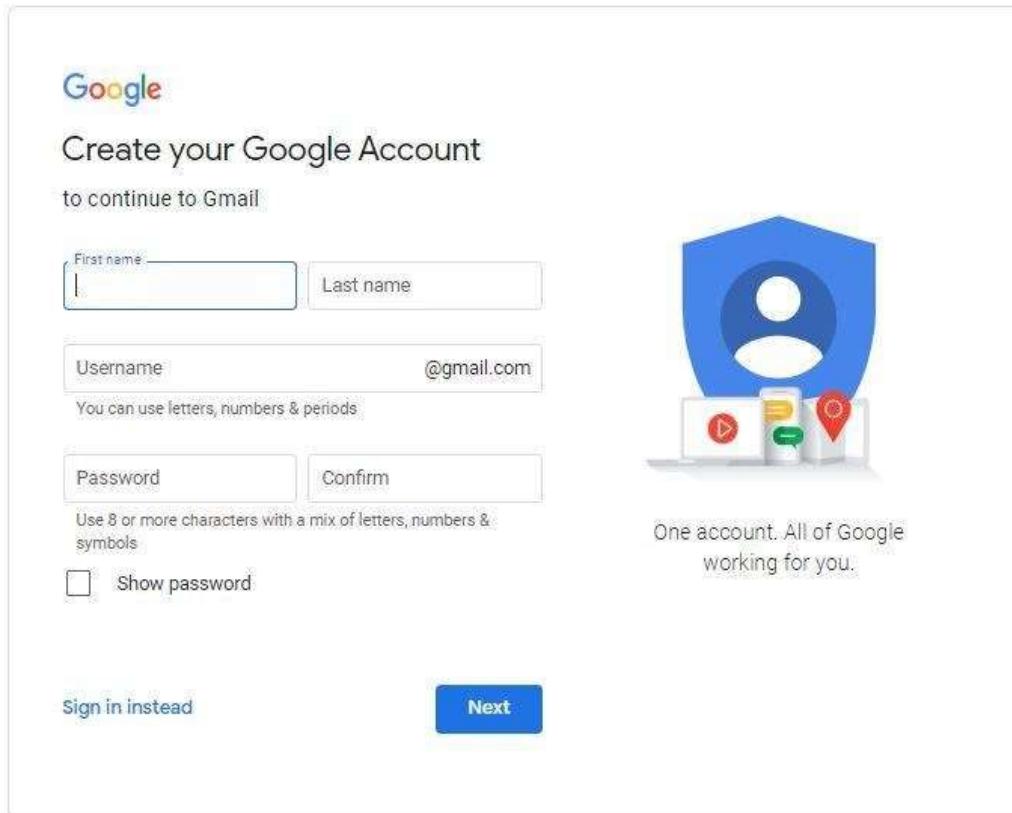
We can take an Example of Gmail Login page. Here is the image of it.

Test Cases for Gmail Login page

The image shows the Gmail sign-in page. At the top center is the Google logo followed by the word "Sign in". Below it is the text "to continue to Gmail". In the center is a large input field labeled "Email or phone" with a placeholder "Email or phone" and a cursor inside. Below the input field is a blue link "Forgot email?". Further down, there is a note "Not your computer? Use Guest mode to sign in privately." followed by a blue link "Learn more". At the bottom left is a blue link "Create account" and at the bottom right is a blue button labeled "Next". At the very bottom of the page, there are small links for "English (United States) ▾", "Help", "Privacy", and "Terms".

Sr. No.	Test Scenarios
1	Enter the valid email address & click next. Verify if the user gets an option to enter the password
2	Don't enter an email address or phone number & just click the Next button. Verify if the user will get the correct message or if the blank field will get highlighted.
3	Enter the invalid email address & click the Next button. Verify if the user will get the correct message.
4	Enter an invalid phone number & click the Next button. Verify if the user will get the correct message.
5	Verify if a user can log in with a valid email address and password.
6	Verify if a user can log in with a valid phone number and password.
7	Verify if a user cannot log in with a valid phone number and an invalid password.
8	Verify if a user cannot log in with a valid email address and a wrong password.
9	Verify the "Forgot email" functionality.
10	Verify the "Forgot password" functionality.

Test Scenarios for the Sign-up page

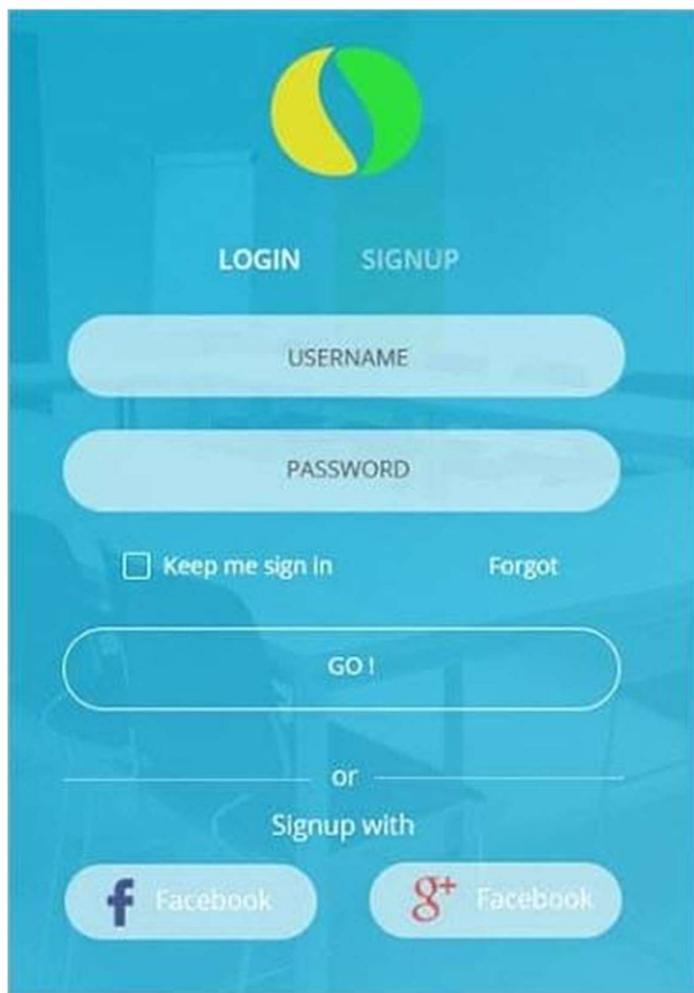


The screenshot shows the 'Create your Google Account' page. At the top, it says 'Create your Google Account' and 'to continue to Gmail'. Below this, there are fields for 'First name' (with a placeholder 'John') and 'Last name' (with a placeholder 'Doe'). There is also a 'Username' field containing '@gmail.com' and a note 'You can use letters, numbers & periods'. Below the username is a 'Password' field with a note 'Use 8 or more characters with a mix of letters, numbers & symbols', and a 'Confirm' field. A checkbox labeled 'Show password' is present. To the right, there is a large blue hexagonal icon with a white person silhouette, and below it, a row of small icons for Google services like YouTube, Google Photos, and Google Maps. A text overlay says 'One account. All of Google working for you.' At the bottom, there are 'Sign in instead' and 'Next' buttons.

Test Scenarios for the Sign-up page

- 1) Verify the messages for each mandatory field.
- 2) Verify if the user cannot proceed without filling all the mandatory fields.
- 3) Verify the age of the user when the DOB is selected.
- 4) Verify if the numbers and special characters are not allowed in the First and Last name.
- 5) Verify if a user can sign-up successfully with all the mandatory details.
- 6) Verify if a user can log in with the valid details.
- 7) Verify if the Password and Confirm Password fields are accepting similar strings only.
- 8) Verify if the Password field will prompt you for the weak passwords.
- 9) Verify if duplicate email address will not get assigned.
- 10) Verify that hints are provided for each field on the form, for the ease of use.

Test Scenarios for the Login page of Mobile Application



- 1) Verify if a user can log in with a valid username and password.
- 2) Verify if a user cannot log in with an invalid username or password. Check permutation and combinations of this.
- 3) Verify the ‘Keep me Sign In’ option. If this check box is selected, then the user should not get logged out even after exiting the app.
- 4) Verify if this check box is not selected by default.
- 5) If the user has signed up with Facebook or social media, verify that the user can log in with those credentials or not.
- 6) Verify the Forgot password functionality.
- 7) Verify if the login page fits the mobile screen. The user should not have to scroll the screen.

Conclusion:

While writing test cases for login or sign-up page write the test cases for all the fields. There should be a combination of both positive and negative test cases. Try to cover the performance, security, and functional scenarios.

The login page is the page with fewer controls, so even though it is looking simple for testing, it should not be considered as an easy task.

Also many a time it is the first impression of an application, so it should be perfect for user interface and usability.

EXPERIMENT No: 2

Title: Test Scenario for Gmail – Inbox Functionality

Test Case for Gmail: Inbox Functionality

- 1) Verify that a newly received email is displayed as highlighted in the Inbox section.
- 2) Verify that a newly received email has correctly displayed sender email Id or name, mail subject and mail body (trimmed to a single line).
- 3) Verify that on clicking the newly received email, the user is navigated to email content.
- 4) Verify that the email contents are correctly displayed with the desired source formatting.
- 5) Verify that any attachments are attached to the email and are downloadable.
- 6) Verify that the attachments are scanned for viruses before download.
- 7) Verify that all the emails marked as read are not highlighted.
- 8) Verify that all the emails read as well as unread have a mail read time appended at the end on the email list displayed in the inbox section.
- 9) Verify that count of unread emails is displayed alongside ‘Inbox’ text in the left sidebar of Gmail.
- 10) Verify that unread email count increases by one on receiving a new email.
- 11) Verify that unread email count decreases by one on reading an email (marking an email as read).
- 12) Verify that email recipients in cc are visible to all users.
- 13) Verify that email recipients in bcc are not visible to the user.
- 14) Verify that all received emails get piled up in the ‘Inbox’ section and get deleted in cyclic fashion based on the size availability.
- 15) Verify that email can be received from non-Gmail email Ids like – yahoo, Hotmail etc.

Test Cases for GMail – Compose Mail Functionality

- 1) Verify that on clicking ‘Compose’ button, a frame to compose a mail gets displayed.
- 2) Verify that user can enter email Ids in ‘To’, ‘cc’ and ‘bcc’ sections and also user will get suggestions while typing the emailIds based on the existing emailIds in user’s email list.
- 3) Verify that the user can enter multiple comma-separated emailIds in ‘To’, ‘cc’ and ‘bcc’ sections.
- 4) Verify that the user can type Subject line in the ‘Subject’ textbox.
- 5) Verify that the user can type the email in the email-body section.
- 6) Verify that users can format mail using editor-options provided like choosing font-family, font-size, bold-italic-underline, etc.
- 7) Verify that the user can attach file as an attachment to the email.
- 8) Verify that the user can add images in the email and select the size for the same.
- 9) Verify that after entering emailIds in either of the ‘To’, ‘cc’ and ‘bcc’ sections, entering Subject line and mail body and clicking ‘Send’ button, mail gets delivered to intended receivers.
- 10) Verify that sent mails can be found in ‘Sent Mail’ sections of the sender.
- 11) Verify that mail can be sent to non-gmail emailIds also.
- 12) Verify that all sent emails get piled up in the ‘Sent Mail’ section and get deleted in cyclic fashion based on the size availability.
- 13) Verify that the emails composed but not sent remain in the draft section.
- 14) Verify the maximum number of email recipients that can be entered in ‘To’, ‘cc’ and ‘bcc’ sections.
- 15) Verify the maximum length of text that can be entered in the ‘Subject’ textbox.
- 16) Verify the content limit of text/images that can be entered and successfully delivered as mail body.
- 17) Verify the maximum size and number of attachments that can be attached with an email.
- 18) Verify that only the allowed specifications of the attachment can be attached with an email.
- 19) Verify that if the email is sent without Subject, a pop-up is generated warning user about no subject line. Also,
- 20) Verify that on accepting the pop-up message, the user is able to send the email.

EXPERIMENT No: 3

Title: Write Test cases in excel sheet for Social Media application or website.

Sample Test Case Template with Test Case Examples:

Every day I keep on getting several requests for a Test Case Template. I'm surprised that many testers are still documenting test cases with Word docs or Excel files.

Most of them prefer excel spreadsheets because they can easily group test cases by test types and most importantly, they can easily get test metrics with Excel formulas. But I'm sure that as the volume of your tests goes on increasing, you will find it extremely difficult to manage.

If you are not using any Test case management tool, then I would strongly recommend you to use an open-source tool to manage and execute your test cases.

Test case formats may vary from one organization to another. However, using a standard test case format for writing test cases is one step closer to setting up a testing process for your project.

It also minimizes Ad-hoc testing that is done without proper test case documentation. But even if you use standard templates, you need to set up test cases writing, review & approve, test execution and most importantly test report preparation process, etc. by using manual methods.

Also, if you have a process to review the test cases by the business team, then you must format these test cases in a template that is agreed by both the parties.

Recommended Tools

Before continuing with the Test case writing process, we recommend downloading these Test case management tools. This will ease your test plan and test case writing process mentioned in this tutorial.

1) TestRail: TestRail is a web-based tool for test cases and test management. It helps QA and development teams with the efficient management of test cases, plans, and runs. It gives centralized test management, powerful reports & metrics, and increased productivity. It is a scalable and customizable solution. It can be used by small as well as large teams.

Features:

- 1) TestRail makes tracking test results easier.
 - 2) It seamlessly gets integrated with bug trackers, automated tests, etc.
 - 3) Personalized to-do lists, filters, and email notifications will help with boosting productivity.
 - 4) Dashboards and activity reports are for easy tracking and following the status of individual tests, milestones, and projects.
- 2) Katalon Studio:** Katalon Studio is an all-in-one, simple automation tool for web, API, mobile, and desktop trusted by over 850,000 users. It simplifies automation for those without a coding background to create automation test cases from manual tests' steps, a rich library of project templates, record & playback, and a friendly UI.
- 3) Testiny:** Testiny – a new, straightforward test management tool, but much more than just a slimmed-down app. Testiny is a fast-growing web application built on the latest technologies and aims to make manual testing and QA management as seamless as possible. It is designed to be extremely easy to use. It helps testers perform tests without adding bulky overhead to the testing process.

Don't just take our word for it, take a look at Testiny yourself. Testiny is perfect for small to mid-sized QA teams looking to integrate manual and automated testing into their development process.

Features:

- 1) Free for open-source projects and small teams with up to 3 people.
- 2) Intuitive and simple out of the box.
- 3) Easily create and handle your test cases, test runs, etc.
- 4) Powerful integrations (e.g. Jira, ...)
- 5) Seamless integration in the development process (linking requirements and defects)
- 6) Instant updates – all browser sessions stay in sync.
- 7) Immediately see if a colleague has made changes, completed a test, etc.
- 8) Powerful REST API.

- 9) Organize your tests in a tree structure – intuitive and easy.

Standard Fields of a Sample Test Case Template:

There are certain standard fields that need to be considered while preparing a Test case template.

Project Name:	
Test Case Template	
Test Case ID: Fun_10	Test Designed by: <Name>
Test Priority (Low/Medium/High): Med	Test Designed date: <Date>
Module Name: Google login screen	Test Executed by: <Name>
Module Name: Google login screen	Test Executed by: <Name>
Test Title: Verify login with valid username and password	Test Execution date: <Date>
Description: Test the Google login page	
Pre-condition: User has valid username and password	
Dependencies:	

Several standard fields for a sample Test Case template are listed below.

- 1) Test case ID: Unique ID is required for each test case. Follow some conventions to indicate the types of the test. For Example, ‘TC_UI_1’ indicating ‘user interface test case #1’.
- 2) Test priority (Low/Medium/High): This is very useful during test execution. Test priorities for business rules and functional test cases can be medium or higher, whereas minor user interface cases can be of a low priority. Testing priorities should always be set by the reviewer.
- 3) Module Name: Mention the name of the main module or the sub-module.
- 4) Test Designed By: Name of the Tester.
- 5) Test Designed Date: Date when it was written.
- 6) Test Executed By Name of the Tester who executed this test. To be filled only after test execution.
- 7) Test Execution Date: Date when the test was executed.
- 8) Test Title/Name: Test case title. For example, verify the login page with a valid username and password.
- 9) Test Summary/Description: Describe the test objective in brief.

- 9) Pre-conditions: Any prerequisite that must be fulfilled before the execution of this test case. List all the pre-conditions in order to execute this test case successfully.
- 10) Dependencies: Mention any dependencies on other test cases or test requirements.
- 11) Test Steps: List all the test execution steps in detail. Write test steps in the order in which they should be executed. Make sure to provide as many details as you can.
- 12) Test Data: Use of test data as an input for this test case. You can provide different data sets with exact values to be used as an input.
- 13) Expected Result: What should be the system output after test execution? Describe the expected result in detail including the message/error that should be displayed on the screen.
- 14) Post-condition: What should be the state of the system after executing this test case?
- 15) Actual result: The actual test result should be filled after test execution. Describe the system behavior after test execution.
- 16) Status (Pass/Fail): If the actual result is not as per the expected result, then mark this test as failed. Otherwise, update it as passed.
- 17) Notes/Comments/Questions: If there are any special conditions to support the above fields, which can't be described above or if there are any questions related to expected or actual results then mention them here.

Add the following fields if necessary:

- 1) Defect ID/Link: If the test status fails, then include the link to the defect log or mention the defect number.
- 2) Test Type/Keywords: This field can be used to classify tests based on test types. For Example, functional, usability, business rules, etc.
- 3) Requirements: Requirements for which this test case is being written for. Preferably the exact section number in the requirement doc.
- 4) Attachments/References: This field is useful for complex test scenarios in order to explain the test steps or expected results using a Visio diagram as a reference. Provide a link or location to the actual path of the diagram or document.
- 5) Automation? (Yes/No): Whether this test case is automated or not. It is useful to track automation status when test cases are automated.

Example Scenario

Based on the above template, below is an example that showcases the concepts in a more understandable way.

Suppose you are testing the login functionality of any web application, say Facebook. Below are the test cases for the same:

Test Scenario ID	Login -1	Test Case ID	Login-1 A
Test Case Description	Login – Positive test case	Test Priority	High
Pre-Requisite	A valid user account	Post- Requisite	NA

Sr. No.	Action	Inputs	Expected Output	Actual Output	Test Browser	Test Result	Test Comments
1	Launch application	https://www.facebook.com	Facebook Home	Facebook home	IE – 11	Pass	Launch successful
2	Enter correct Email & Password and hit login button	Email id: test@xyz.com Password: *****	Login success	Login success	IE – 11	Pass	Launch successful

Test Scenario ID	Login -1	Test Case ID	Login-1 B
Test Case Description	Login – Negative test case	Test Priority	High
Pre-Requisite	NA	Post- Requisite	NA

Test Execution Steps:							
Sr. No.	Action	Inputs	Expected Output	Actual Output	Test Browser	Test Result	Test Comments
1	Launch	https://www.facebook.com	Facebook	Facebook	IE-11	Pass	Launch Successful
2	Enter invalid Email & any Password and hit login button	Email id: invalid@xyz.com Password: *****	The email address or phone number that you've entered doesn't match any account. Sign up for an account	The email address or phone number that you've entered doesn't match any account. Sign up for an account	IE-11	Pass	Invalid login attempt stopped
3	Enter valid Email & incorrect password and hit login button	Email id: valid@xyz.com Password: *****	The password that you've entered is incorrect. Forgotten password ?	The password that you've entered is incorrect. Forgotten password ?	IE – 11	Pass	Invalid login attempt stopped

Samples from a Live Project

Below is examples from a live project that demonstrates how all the above listed tips and tricks are actually implemented:

1	Test Scenario Group	Test Case Id	Test Case Description	Test Env	Test Input	Expected Result	Actual Result	Test Browser	Executed Date	Test Results	Executed Date	Test Results	Defect Status
2	Search	Search-1	Search for Patient only in Copia										
3	Search	Search-2	Search for Patient only in IMS										
4	Search	Search-3	Search for Patient in both Copia & IMS										
5	Search	Search-4	Search by first name										
6	Search	Search-5	Search by last name										
7	Search	Search-6	Search by patient id										
8	Search	Search-7	Search with partial name										
9	Search	Search-8	Search with partial patient id										
10	Search	Search-9	Update Patient profile a) address flow from HRA to IMS										
11	Search	Search-10	Search recent history										
12	Search	Search-11	Search on different first name & lastname gives lastname										
13			Edit patient profile on search page										
14	HRA	HRA-1	Verify all HRA data flowed to										
15	HRA	HRA-2	IMS										
16													
17													

Summary Bugs | IMS-Search | Interaction | Quick Actions | HRA | Personal Health History ... + : []

Timestamp	Description	Which Page?	Issue Type	Screenshot	Urgency	Browser	Re-test Date	Testing Comments
2/7/13 17:08	[Clicking on enter after entering data closes the modal Steps: a) Click on 'Create New Patient' b) Enter patient name and click enter c) Create patient modal is closed	Login	Error or Bug	None	Medium	Firefox	2/21/2013	Pending
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								

← → | Summary | **Bugs** | IMS-Search | Interaction | Quick Actions | HRA | Personal Health History ... + : [x] |

Test Scenario Group	Test Case Id	Test Case Description	Test Input	Expected Result	Actual Result	Test Browser	Executed Date	Test Results
2 Health records Personal								
3 Health History PH 1 High Cholesterol:		a) Yes and on medication	how old were you at time of dx?	how old were you at time of dx?	Chrome,IE, Firefox			
		b) Yes but not on medication	how old were you at time of dx?	how old were you at time of dx?	Chrome,IE, Firefox			
4			what does your blood pressure range, on average?	what does your blood pressure range, on average?	Chrome,IE, Firefox			
5 PH 2 High Blood Pressure:		a) Yes and on medication	what does your blood pressure range, on average?	what does your blood pressure range, on average?	Chrome,IE, Firefox			
		b) Yes but not on medication	how old were you at time of dx?	how old were you at time of dx?	Chrome,IE, Firefox			
6		a) Yes and on medication	how old were you at time of dx?	how old were you at time of dx?	Chrome,IE, Firefox			
7		b) Yes but not on medication	how old were you at time of dx?	how old were you at time of dx?	Chrome,IE, Firefox			
8		a) Yes and on medication	how old were you at time of dx?	how old were you at time of dx?	Chrome,IE, Firefox			
9 PH 3 Diabetes:		b) Yes but not on medication	what type of dm do you have?	what type of dm do you have?	Chrome,Firefox			
10		a) Yes and on medication	what type of dm do you have?	what type of dm do you have?	Chrome,Firefox			
11		b) Yes but not on medication	what type of dm do you have?	what type of dm do you have?	Chrome,Firefox			
12		a) Yes and on medication	what type of dm do you have?	what type of dm do you have?	Chrome,Firefox			

EXPERIMENT No: 4

Title: Create Defect Report for Any application or web application.

Sample Bug Report:

Sample Bug Report

The Sample, Bug/Defect Report given below will give you an exact idea of how to report a Bug in the Bug Tracking Tool?

Here is an Example scenario that caused a Bug:

Let's assume that in your application under test you want to create a new user with user information, for that you need to login into the application and navigate to the USERS menu -> New User, then enter all the details in the 'User form' like, First Name, Last Name, Age, Address, Phone etc.

Once you enter all this information, you need to click on the 'SAVE' button in order to save the user. Now you can see a successful message saying, "New User has been created successfully".

But when you entered into your application by logging in and you have navigated to the USERS menu -> New user, entered all the required information to create the new user and clicked on SAVE button.

BANG! The application crashed and you got one error page on the screen. (Capture this error message window and save it as a Microsoft paint file)

Now, this is a Bug scenario and you would like to report this as a BUG in your Bug-Tracking Tool. How Will You Report This Bug Effectively?

Sample Bug Report

Here is a sample Bug Report for the above-mentioned example:

(Note that some 'Bug Report' fields might differ depending on your bug tracking system)

SAMPLE BUG REPORT

Bug Name: Application crashes upon clicking the SAVE button while creating a new user.

Bug ID: (It will be automatically created by the BUG Tracking tool once you save this bug).

Area Path: USERS menu -> New Users

Build Number: Version Number 5.0.1 **Severity:** HIGH (High/Medium/Low) or 1 **Priority:**

HIGH (High/Medium/Low) or 1 **Assigned to:** Developer-X

Reported By: Your Name **Reported On:** Date **Reason:** Defect

Status: New/Open/Active (Depends on the Tool you are using) **Environment:** Windows 2003/SQL Server 2005

Description: Application crashes upon clicking the SAVE button while creating a new user, hence unable to create a new user in the application.

Steps to Reproduce:

1) Login into the Application.

2) Navigate to the Users Menu -> New User 3) Filled out all the user information fields. 4)
Clicked on the ‘Save’ button.

5) Seen an error page “ORA1090 Exception: Insert values Error...”

6) See the attached logs for more information (Attach more logs related to the bug..IF any) 7)
Also see the attached screenshot of the error page.

Expected Result: On clicking the SAVE button, you should be prompted to a successful message “New User has been created successfully”.

(Attach ‘application crash’ screenshot. IF any)

Save the Defect/Bug in the BUG TRACKING TOOL. You will get a Bug ID that you can use for further bug reference.

Default ‘New Bug’ mail will go to the respective developer and the default module owner (Team leader or manager) for further action.

How To Write A Good Bug Report? Tips And Tricks

If your Bug report is effective, then its chances of getting fixed are higher. So, fixing a bug depends upon how effectively you report it. Reporting a bug is nothing but a skill and, in this tutorial, we will explain how to achieve this skill.

“The point of writing a problem report (bug report) is to get bugs fixed” – By Cem Kaner. If a tester is not reporting a bug correctly, then the programmer will most likely reject this bug stating it as irreproducible.

This can hurt the tester’s morals and sometimes the ego too. (I suggest not to keep any type of ego. ego’s like “I have reported the bug correctly”, “I can reproduce it”, “Why has he/she rejected the bug?”, “It’s not my fault” etc.,).

Qualities of a Good Software Bug Report

Anyone can write a Bug report. But not everyone can write an effective Bug report. You should be able to distinguish between an average bug report and a good bug report.

How to distinguish between a good and bad Bug Report? It’s very simple, apply the following characteristics and techniques to report a bug.

Characteristics and Techniques

#1) Having a clearly specified Bug Number: Always assign a unique number to each bug report. This, in turn, will help you identify the bug record. If you are using any automated bug-reporting tool then this unique number will be generated automatically each time you report a bug.

Note the number and a brief description of each bug that you reported.

#2) Reproducible: If your bug is not reproducible, then it will never get fixed.

You should clearly mention the steps to reproduce the bug. Do not assume or skip any reproducing steps. The bug which is described Step by step is easy to reproduce and fix.

#3) Be Specific: Do not write an essay about the problem.

Be Specific and to the point. Try to summarize the problem in minimum words yet in an effective way. Do not combine multiple problems even if they seem to be similar. Write different reports for each problem. **Effective Bug Reporting**

Bug reporting is an important aspect of Software Testing. Effective Bug reports communicate well with the development team to avoid confusion or miscommunication.

Good Bug report should be **clear and concise** without any missing key points. Any lack of clarity leads to misunderstanding and slows down the development process as well. Defect writing and reporting is one of the most important but neglected areas in the testing life cycle.

Good writing is very important for filing a bug. The most important point that a tester should keep in mind

is **not to use a commanding tone** in the report. This breaks morale and creates an unhealthy work relationship. Use a suggestive tone.

Don't assume that the developer has made a mistake and hence you can use harsh words. Before reporting, it is equally important to check if the same bug has been reported or not.

A duplicate bug is a burden in the testing cycle. Check out the whole list of known bugs. At times, the developers may be aware of the issue and ignore it for future releases. Tools like Bugzilla, which automatically searches for duplicate bugs, can also be used. However, it is best to manually search for any duplicate bug.

The import information that a bug report must communicate is **“How?” and “Where?”** The report should clearly answer exactly how the test was performed and where the defect occurred. The reader should easily reproduce the bug and find out where the bug is.

Keep in mind that the **objective of writing a Bug report** is to enable the developer to visualize the problem. He/She should clearly understand the defect from the Bug report. Remember to provide all the relevant information that the developer is seeking.

Also, bear in mind that a bug report would be preserved for future use and should be well written with the required information. **Use meaningful sentences and simple words** to describe your bugs. Don't use confusing statements that waste the time of the reviewer.

Report each bug as a separate issue. In case of multiple issues in a single Bug report, you can't close it unless all the issues are resolved.

Hence, it is best to **split the issues into separate bugs**. This ensures that each bug can be handled separately. A well-written bug report helps a developer to reproduce the bug at their terminal. This will help them diagnose the issue as well.

How To Report A Bug?

Use the following simple Bug report template:

This is a simple Bug report format. It may vary depending upon the Bug report tool that you are using. If you are writing a bug report manually then some fields need to be mentioned specifically like the Bug number – which should be assigned manually.

Reporter: Your name and email address. **Product:** In which product you found this bug.

Version: The product version, if any.

Component: These are the major sub-modules of the product.

Platform: Mention the hardware platform where you found this bug. The various platforms like ‘PC’, ‘MAC’, ‘HP’, ‘Sun’ etc.

Operating system: Mention all the operating systems where you found the bug. Operating systems like Windows, Linux, Unix, SunOS, and Mac OS. Also, mention the different OS versions like Windows NT, Windows 2000, Windows XP etc, if applicable.

Priority: When should a bug be fixed? Priority is generally set from P1 to P5. P1 as “fix the bug with the highest priority” and P5 as ” Fix when time permits”.

Severity: This describes the impact of the bug. **Types of Severity:**

- **Blocker:** No further testing work can be done. • **Critical:** Application crash, Loss of data.
- **Major:** Major loss of function. • **Minor:** Minor loss of function.
- **Trivial:** Some UI enhancements.
- **Enhancement:** Request for a new feature or some enhancement in the existing one.

Status: When you are logging the bug into any bug tracking system then by default the bug status will be ‘New’.

Later on, the bug goes through various stages like Fixed, Verified, Reopen, Won’t Fix, etc.

Assign To: If you know which developer is responsible for that particular module in which the bug occurred, then you can specify the email address of that developer. Else keep it blank as this will assign the bug to the module owner, if not the Manager will assign the bug to the developer. Possibly add the manager’s email address to the CC list.

URL: The page URL on which the bug occurred.

Summary: A brief summary of the bug, mostly within 60 words or below.

Make sure your summary is reflecting on what the problem is and where it is.

Description: A detailed description of the bug. **Use the following fields for the description field:**

- **Reproduce steps:** Clearly, mention the steps to reproduce the bug.
- **Expected result:** How the application should behave on the above-mentioned steps.
- **Actual result:** What is the actual result of running the above steps i.e. the bug behavior.

These are the important steps in the bug report. You can also add “Report Type” as one more field which will describe the bug type.

Report Types include: 1) Coding error

2) Design error

3) New Suggestion

4) Documentation issue 5) Hardware problem

Important Features in Your Bug Report

Given below are the important features in the Bug report:

#1) Bug Number/id

A Bug number or an identification number (like swb001) makes bug reporting and the process of referring to bugs much easier. The developer can easily check if a particular bug has been fixed or not. It makes the whole testing and retesting process smoother and easier.

#2) Bug Title

Bug titles are read more often than any other part of the bug report. This should explain all about what comes in the bug. The Bug title should be suggestive enough that the reader can understand it. A clear bug title makes it easy to understand and the reader can know if the bug has been reported earlier or has been fixed.

#3) Priority

Based on the severity of the bug, a priority can be set for it. A bug can be a Blocker, Critical, Major, Minor, Trivial, or a suggestion. Bug priorities can be given from P1 to P5 so that the important ones are viewed first.

#4) Platform/Environment

OS and browser configuration is necessary for a clear bug report. It is the best way to communicate how the bug can be reproduced.

Without the exact platform or environment, the application may behave differently and the bug at the tester's end may not replicate on the developer's end. So it is best to clearly mention the environment in which the bug was detected.

#5) Description

Bug description helps the developer to understand the bug. It describes the problem encountered. A poor description will create confusion and waste the time of the developers as well as testers.

It is necessary to communicate clearly about the effect of the description. It's always helpful to use complete sentences. It is a good practice to describe each problem separately instead of crumbling them altogether. Don't use terms like "I think" or "I believe".

#6) Steps to Reproduce

A good Bug report should clearly mention the steps to reproduce. These steps should include actions that may cause the bug. Don't make generic statements. Be specific on the steps to follow.

A good example of a well-written procedure is given below Steps:

- Select product Abc01.
- Click on Add to cart.
- Click Remove to remove the product from the cart.

#7) Expected and Actual Result

A Bug description is incomplete without the Expected and Actual results. It is necessary to outline what the outcome of the test is and what the user should expect. The reader should know what the correct outcome of the test is. Clearly, mention what happened during the test

and what the outcome was.

#8) *Screenshot*

A picture is worth a thousand words. Take a Screenshot of the instance of failure with proper captioning to highlight the defect. Highlight unexpected error messages with light red color. This draws attention to the required area.

Some Bonus Tips To Write A Good Bug Report

Given below are some additional tips on how to write a good Bug report: **#1) Report the problem immediately**

If you find any bugs while testing, then you do not need to wait to write a detailed bug report later. Instead, write a bug report immediately. This will ensure a good and reproducible Bug report. If you decide to write the Bug report later on then there is a higher chance to miss the important steps in your report.

#2) Reproduce the bug three times before writing a Bug report

Your bug should be reproducible. Make sure that your steps are robust enough to reproduce the bug without any ambiguity. If your bug is not reproducible every time, then you can still file a bug mentioning the periodic nature of the bug.

#3) Test the same bug occurrence on other similar modules

Sometimes the developer uses the same code for different similar modules. So there is a higher chance for the bug in one module to occur in other similar modules as well. You can even try to find the more severe version of the bug you found.

#4) Write a good bug summary

Bug summary will help the developers to quickly analyze the bug's nature. A poor quality report will unnecessarily increase development and testing time. Communicate well with your bug report summary. Keep in mind that the bug summary can be used as a reference to search for the bug in the bug inventory.

#5) Read the Bug report before hitting the Submit button

Read all the sentences, wordings and steps that are used in the bug report. See if any sentence is creating ambiguity that can lead to misinterpretation. Misleading words or sentences should be avoided in order to have a clear bug report.

#6) Do not use abusive language.

It's nice that you did a good work and found a bug but do not use this credit for criticizing the developer or to attack any individual.

Sample Bug Reports for Web and Product Applications

Bug report sample 1: Web Project bug report

Summary: In CTR (Click through ratio) “Total” row calculation is wrong

Product: Example product

Version: 1.0

Platform: PC

URL: (Provide url of page where bug occurs)

OS/Version: Windows 2000

Status: NEW

Severity: Major

Priority: P1

Component: Publisher stats

Assigned To: developer@example.com

Reported By: tester@example.com

CC: manager@example.com

Bug Description:

Reproduce steps:

- 1) Go to page: (Provide URL of page where bug occurs)
- 2) Click on ‘Publisher stats’ link to view publisher’s revenue detail stats date wise.
- 3) On page (Provide URL of page where bug occurs) check CTR value in ‘Total’ row of CTR stats table. Actual result: Calculation of ‘Total’ row in CTR table is wrong. Also, Individual row CTR for each publisher is not truncated to 2 digits after decimal point. It’s showing CTR like 0.042556767

Expected result: Total CTR= (Total clicks/Total searches) *100 [Attach bug screenshot if any]

Please fix the bug.

Sample bug report 2: Application product Bug report sample Application testing scenario:

Lets assume in your application you want to create a new user with his/her information, for that you need to logon into the application and navigate to USERS menu > New User, then enter all the details in the User form like, First Name, Last Name, Age, Address, Phone etc. Once you enter all these need to click on SAVE button in order to save the user and you can see a success message saying, "New User has been created successfully". Now you entered into your application by logging in and navigate to USERS menu > New user, entered all the information and clicked on SAVE button and now the application crashed and you can see one error page on the screen, now you would like to report this BUG.

Now here is how we can report bug for above scenario:

Bug Name: Application crash on clicking the SAVE button while creating a new user. **Bug ID:** The BUG Tracking tool will automatically create it once you save this.

Area Path: USERS menu > New Users **Build Number:**/Version Number 5.0.1 **Severity:** HIGH (High/Medium/Low) **Priority:** HIGH (High/Medium/Low) **Assigned to:** Developer-X

Created By: Your Name **Created On:** Date **Reason:** Defect

Status: New/Open/Active – Depends on

the Tool you are using **Environment:**

Windows 2003/SQL Server 2005

Description:

Application crash on clicking the SAVE button while creating a new user, hence unable to create a new user in the application.

Steps To Reproduce:

- 1) Logon into the application
- 2) Navigate to the Users Menu > New User
- 3) Filled all the fields

- 4) Clicked on ‘Save’ button
- 5) Seen an error page “ORA1090 Exception: Insert values Error...”
- 6) See the attached logs for more information
- 7) And also see the attached screenshot of the error page.

Expected: On clicking SAVE button should be prompted to a success message “New User has been created successfully”.

Save the defect/bug in the BUG TRACKING TOOL.

EXPERIMENT No: 5

Title: Installation of Selenium grid and selenium Web driver java eclipse (Automation tools).

Managing user experience is pivotal for software development. Test automation enables user preferences and

convenience to remain at the center of the development process while saving time and effort. That is why comprehensive automation testing has become necessary to retain customers and meet their expectations. With significantly shorter time frames for development, Selenium Testing, in particular, has become an integral part of the development to facilitate automated testing of web applications.

Selenium is the most popular automated tool in existence today. 59.5% of people consider Selenium for Cross Browser Testing because of the robustness and flexibility it offers by supporting multiple languages like Java, C#, Python, Perl, Ruby, etc. However, a majority (67%) of the Selenium users prefer Java as their language for Selenium Testing.

This article discusses how to configure Selenium in Eclipse to use Selenium for Java. Table of Contents

- Prerequisites for configuring Selenium in Eclipse • How to configure Selenium in Eclipse
- Conclusion

Prerequisites for configuring Selenium in Eclipse

Install JavaDownload Java SE Development Kit 16.0.2 according to the Windows, Linux, or macOS platform being used.

Product / File Description	File Size	Download
Linux ARM 64 RPM Package	144.87 MB	 jdk-16.0.2_linux-aarch64_bin.rpm
Linux ARM 64 Compressed Archive	160.73 MB	 jdk-16.0.2_linux-aarch64_bin.tar.gz
Linux x64 Debian Package	146.17 MB	 jdk-16.0.2_linux-x64_bin.deb
Linux x64 RPM Package	153.01 MB	 jdk-16.0.2_linux-x64_bin.rpm
Linux x64 Compressed Archive	170.04 MB	 jdk-16.0.2_linux-x64_bin.tar.gz
macOS Installer	166.6 MB	 jdk-16.0.2_osx-x64_bin.dmg
macOS Compressed Archive	167.21MB	 jdk-16.0.2_osx-x64_bin.tar.gz
Windows x64 Installer	150.58 MB	 jdk-16.0.2_windows-x64_bin.exe
Windows x64 Compressed Archive	168.8 MB	 jdk-16.0.2_windows-x64_bin.zip

Source

Run the JDK Installer by double-clicking on the file name in the download location and following the instructions on the instruction wizard. Alternatively, silently install JDK by entering the following command: `jdk.exe /s`

Install Eclipse IDE

The Eclipse Installer 2021-06 R now includes a JRE for macOS, Windows and Linux.

Get **Eclipse IDE 2021-06**

Install your favorite desktop IDE packages.

[Download x86_64](#)

[Download Packages](#) | [Need Help?](#)

Tool Platforms

Eclipse Che

Eclipse Che is a developer workspace server and cloud IDE.

Orion

A modern, open source software development environment that runs in the cloud.

Install Selenium

Download and Install Selenium to be set up in Eclipse.

Install Browser Driver

For Cross Browser Testing, download the relevant Browser Driver – ChromeDriver (for Chrome), GeckoDriver (for Firefox), SafariDriver(for Safari), and InternetExplorerDriver and MSEdgeDriver (IE and Edge respectively). Place these Browser Driver files in a directory that is part of the environment PATH. This will allow a command-line call to the programs to execute them irrespective of the working directory.

Install Java Language Bindings

Version 3.141.59 (2018)

[Changelog](#)

[API Docs](#)

The screenshot shows the Selenium website's language bindings section. It features five cards, each representing a programming language:

- C#**: Shows the C# logo (a purple hexagon with a white 'C#' symbol). Below it, it says "Stable: 3.14.0 (August 02, 2018)" and "Beta: 4.0.0-beta4 (June 07, 2021)". Underneath are links for [Changelog](#) and [API Docs](#).
- Ruby**: Shows the Ruby logo (a red gem-like icon). Below it, it says "Stable: 3.142.6 (October 04, 2019)" and "Beta: 4.0.0beta4 (June 07, 2021)". Underneath are links for [Changelog](#) and [API Docs](#).
- Java**: Shows the Java logo (a steaming coffee cup). Below it, it says "Stable: 3.141.59 (November 14, 2018)" and "Beta: 4.0.0-beta-4 (June 07, 2021)". Underneath are links for [Changelog](#) and [API Docs](#).
- Python**: Shows the Python logo (two interlocking snakes). Below it, it says "Stable: 3.141.0 (November 01, 2018)" and "Beta: 4.0.0.b4 (June 07, 2021)". Underneath are links for [Changelog](#) and [API Docs](#).
- JavaScript**: Shows the JS logo (a yellow square with the letters 'JS'). Below it, it says "Stable: 3.6.0 (October 06, 2017)" and "Beta: 4.0.0-beta.4 (June 07, 2021)". Underneath are links for [Changelog](#) and [API Docs](#).

How to configure Selenium in Eclipse

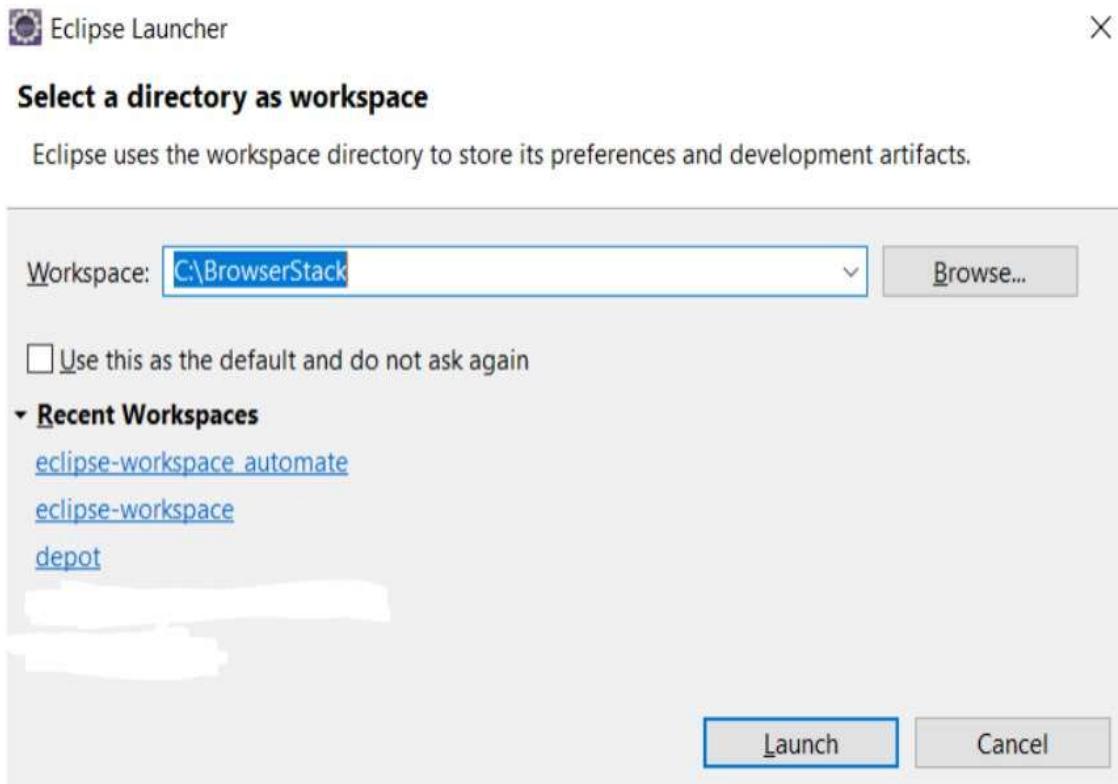
Here are the steps to configure Selenium Webdriver with Eclipse:

Step 1: Launch Eclipse

To launch Eclipse double click on the `eclipse.exe` file in the download location.

Step 2: Create Workspace in Eclipse

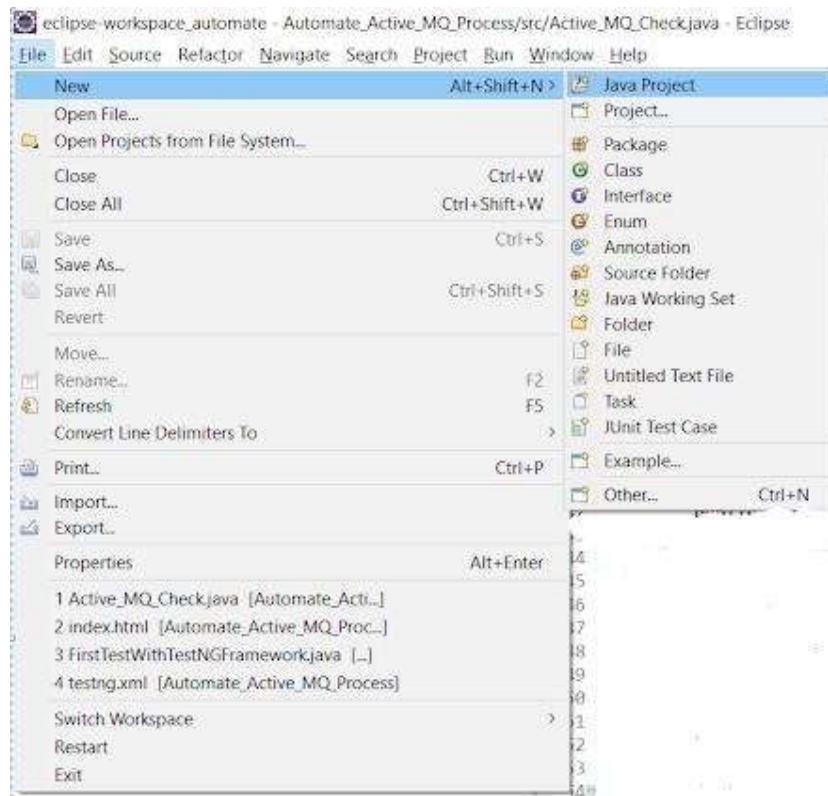
This workspace named “`C:\BrowserStack`” is like any other folder, which will store all the test scripts. Launch the BrowserStack workspace.



Creating Workspace in Eclipse

Step 3: Create New Java Project in the BrowserStack Workspace

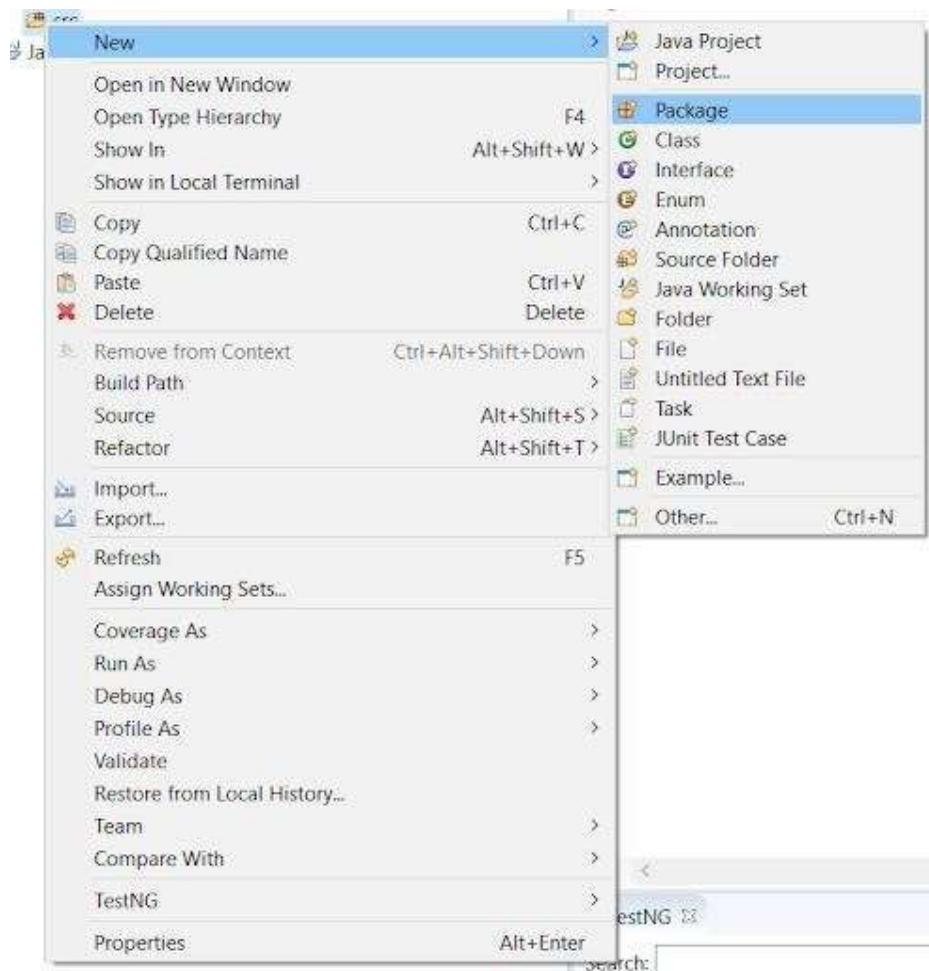
Create a new Java Project by clicking on File > New > Java Project and name it.



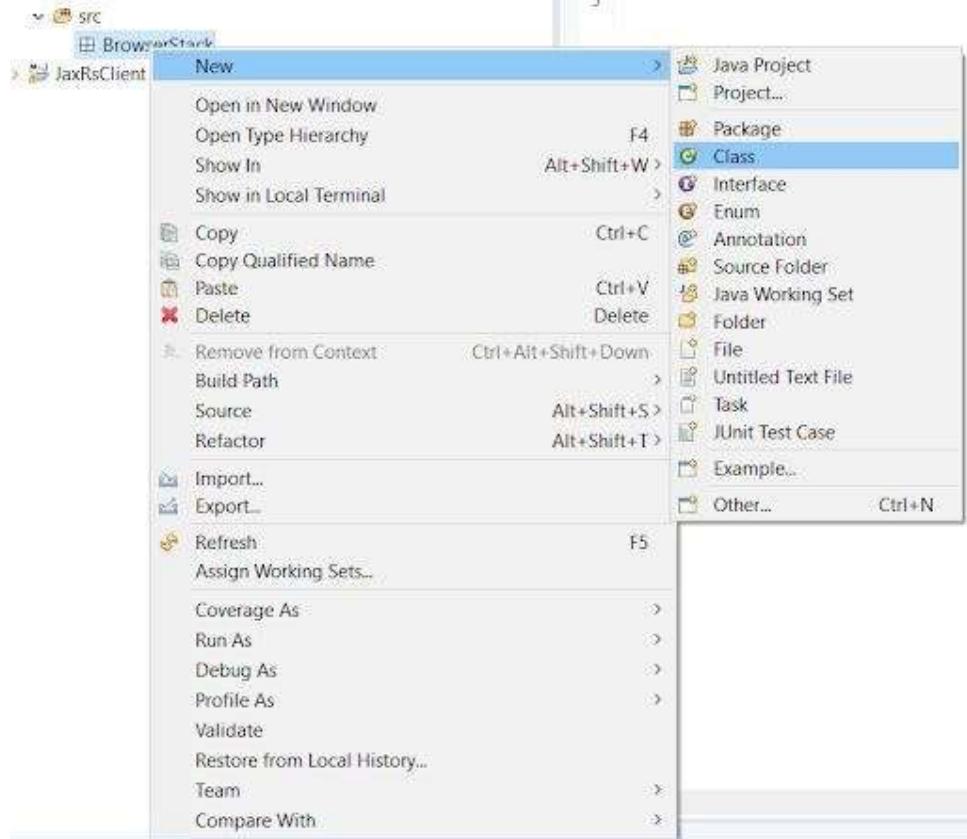
Creating a new Java Project

Step 4: Create Package and Class under the Java Project

By clicking on the src folder (which is the source folder), create a new package and name it (BrowserStack). Then right-click on the package name and create a class.



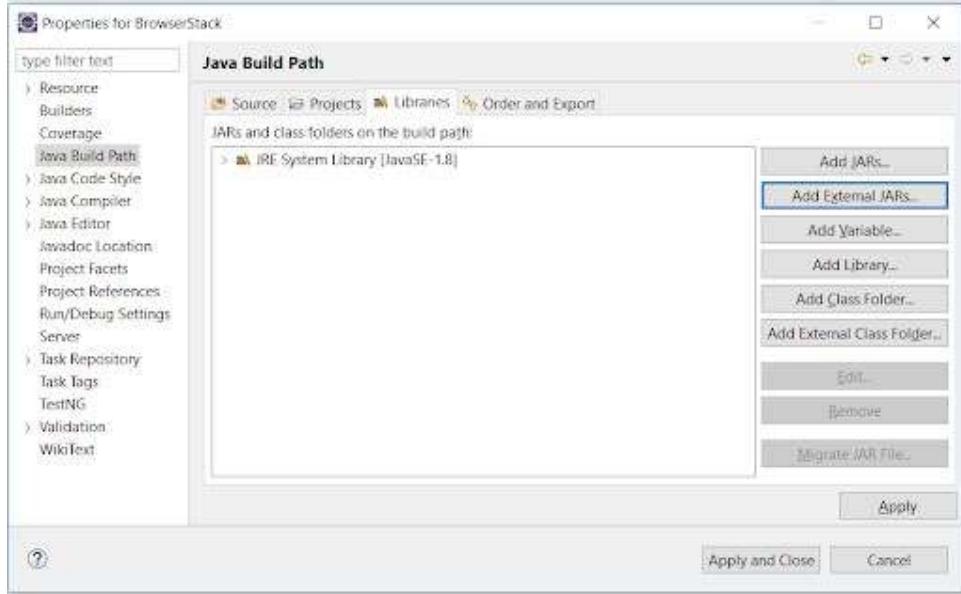
Creating Package in the Java Project



Creating Class in the BrowserStack Package

Step 5: Add Selenium JARs to the Java Project in Eclipse

To add the Selenium Jars to the BrowserStack Java right click on the BrowserStack Project folder and select the Properties option. In the properties window, click on the Java Build Path and Add External JARs. Browse and add the downloaded Selenium JARs i.e. Client Combined JAR and all the JARs under the Libs folder, then click Apply and Close.



Adding Selenium JARs in the BrowserStack Project

This configures Selenium with Eclipse, making it ready to execute the first test script.

Conclusion:

Automation testing using Selenium with Java has simplified software development. Being an open-source tool, it provides an opportunity to speed up the time of execution and remove errors for a better user experience.

Java is quite popular among the developers, given the extensive support being available, making it a preferred choice as a Selenium Client Language Binding. Selenium with Java has proved helpful in optimizing regression testing and cross browser testing. With the support of Continuous Integration tools like Jenkins, Selenium with Java can be used in continuous delivery models.

However, to make testing more efficient and get accurate results every time, all Selenium tests must be run on real browsers and devices for accurate results. Start running tests on 2000+ real browsers and devices on BrowserStack's real device cloud. Run parallel tests on its Cloud Selenium Grid to get faster results without compromising on accuracy. Identify all bugs and offer a high-end UX/UI to the users by running automated tests in real user conditions with BrowserStack Automate.

Mini Project No.,1

1.1 Title

Mini-Project 1: Create a small application by selecting relevant system environment/ platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Prepare Test Cases inclusive of Test Procedures for identified Test Scenarios. Perform selective Black-box and White-box testing covering Unit and Integration test by using suitable Testing tools. Prepare Test Reports based on Test Pass/Fail Criteria and judge the acceptance of application developed.

1.2 Problem Definition:

Perform Desktop Application testing using Automation Tool like JUnit generate Test Report by Using tool like Apache Maven.

1.3 Prerequisite:

Knowledge of Core Java, Basic Concepts of Unit Testing, Test Cases Writing using Junit etc tool

1.4 Software Requirements:

JDK 1.8, Eclipse java photon-R version, TestNG

1.5 Hardware Requirement:

PIV, 2GB RAM, 500 GB HDD, Lenovo A13-4089Model.

1.6 Learning Objectives:

We are going to learn how to Prepare Test Cases inclusive of Test Procedures for identified Test Scenarios. Perform selective Black-box and White-box testing covering Unit and Integration test by using suitable Testing tools. also Prepare Test Reports based on Test Pass/Fail Criteria

1.7 Outcomes:

You are able to understand Unit and Integration testing with Tool with Test Report. **1.8**

Theory Concepts:

1.8.1 What is Unit Testing?

Unit Testing of software applications is done during the development (coding) of an application.

The objective of Unit Testing is to isolate a section of code and verify its correctness.

In procedural programming a unit may be an individual function or procedure

The goal of Unit Testing is to isolate each part of the program and show that the individual parts are correct. Unit Testing is usually performed by the developer.

1.8.2 Unit Testing Tools

There are several automated tools available to assist with unit testing. We will provide a few examples below:

1. [Jtest](#): Parasoft Jtest is an IDE plugin that leverages open-source frameworks (JUnit, Mockito, PowerMock, and Spring) with guided and easy one-click actions for creating, scaling, and maintaining unit tests. By automating these time-consuming aspects of unit testing, it frees the developer to focus on business logic and create more meaningful test suites.
2. [Junit](#): Junit is a free to use testing tool used for Java programming language. It provides assertions to identify test method. This tool test data first and then inserted in the piece of code.
3. [NUnit](#): NUnit is widely used unit-testing framework use for all .net languages. It is open source tool which allows writing scripts manually. It supports data-driven tests which can run in parallel.
4. [JMockit](#): JMockit is open source Unit testing tool. It is code coverage tool with line and path metrics. It allows mocking API with recording and verification syntax. This tool offers Line coverage, Path Coverage, and Data Coverage.
5. [EMMA](#): EMMA is an open-source toolkit for analyzing and reporting code written in Java language. Emma support coverage types like method, line, basic block. It is Java-based so it is without external library dependencies and can access to the source code.

6. [PHPUnit](#): PHPUnit is a unit testing tool for PHP programmer. It takes small portions of code which is called units and test each of them separately. The tool also allows developers to use pre-define assertion methods to assert that system behave in a certain manner.

Those are just a few of the available unit testing tools. There are lots more, especially for C languages and Java, but you are sure to find a unit testing tool for your programming needs regardless of the language you use.

1.8.3 Extreme Programming & Unit Testing

Unit testing in Extreme Programming involves the extensive use of testing frameworks. A unit test framework is used in order to create automated unit tests. Unit testing frameworks are not unique to extreme programming, but they are essential to it. Below we look at some of what extreme programming brings to the world of unit testing:

- Tests are written before the code
- Rely heavily on testing frameworks
- All classes in the applications are tested
- Quick and easy integration is made possible

1.8.4 Bug taxonomy

Bug taxonomies help in providing fast and effective feedback so that they can easily identify possible reasons for failure of the software. Using bug taxonomy, a large number of potential bugs can be grouped into few categories.

Whenever a new bug is reported, using bug taxonomy, a tester can easily analyse and put that bug into any of these categories.

At the end of testing, Testers can understand the type of categories of bugs that frequently occurred and thereby in successive rounds of testing he can focus on writing more test cases that would help to detect such bugs. In addition, test leaders can guide their testers to focus on such frequently occurring bugs.

The summary of the Bug Taxonomy is given below,

- Requirements, Features, and Functionality Bugs

- Structural Bugs
- Data Bugs
- Coding Bugs
- Interface, Integration, and System Bugs • Test and Test Design Bugs
- Testing and Design Style **1.8.5 What is Integration Testing?**

In integration Testing, individual software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. integration Testing focuses on checking data communication amongst these modules. Hence it is also termed as 'I & T' (Integration and Testing), 'String Testing' and sometimes 'Thread Testing **Integration Test Case:**

Integration Test Case differs from other test cases in the sense it **focuses mainly on the interfaces & flow of data/information between the modules**. Here priority is to be given for the **integrating links** rather than the unit functions which are already tested.

Sample Integration Test Cases for the following scenario: Application has 3 modules say 'Login Page', 'Mail box' and 'Delete mails' and each of them are integrated logically.

Here do not concentrate much on the Login Page testing as it's already been done in [Unit Testing](#). But check how it's linked to the Mail Box Page.

Similarly Mail Box: Check its integration to the Delete Mails Module.

Test Case ID	Test Case Objective	Test Case Description	Expected Result
1	Check the interface link between the Login and Mailbox module	Enter login credentials and click on the Login button	To be directed to the Mail Box
2	Check the interface link between the Mailbox and Delete Mails Module	From Mail box select the an email and click delete button	Selected email should appear in the Deleted/Trash folder

1.8.6 Desktop Application Testing by Using Junit Tool What is Junit?

JUnit is a framework for implementing testing in Java.

It provides a simple way to explicitly test specific areas of a Java program, it is extensible and can be employed to test a hierarchy of program code either singularly or as

multiple units.

Why use a testing framework? Using a testing framework is beneficial because it forces you to explicitly declare the expected results of specific program execution routes. When debugging it is possible to write a test which expresses the result you are trying to achieve and then debug until the test comes out positive.

By having a set of tests that test all the core components of the project it is possible to modify specific areas of the project and immediately see the effect the modifications have on the other areas by the

results of the test, hence, side-effects can be quickly realized.

JUnit promotes the idea of first testing then coding, in that it is possible to setup test data for a unit which defines what the expected output is and then code until the tests pass. It is believed by some that this practice of "test a little, code a little, test a little, code a little..." increases programmer productivity and stability of program code whilst reducing programmer stress and the time spent debugging.

JUnit is a simple open source Java testing framework used to write and run repeatable automated tests.

It is an instance of the xUnit architecture for unit testing framework. Eclipse supports creating test cases and running test suites, so it is easy to use for your Java applications.

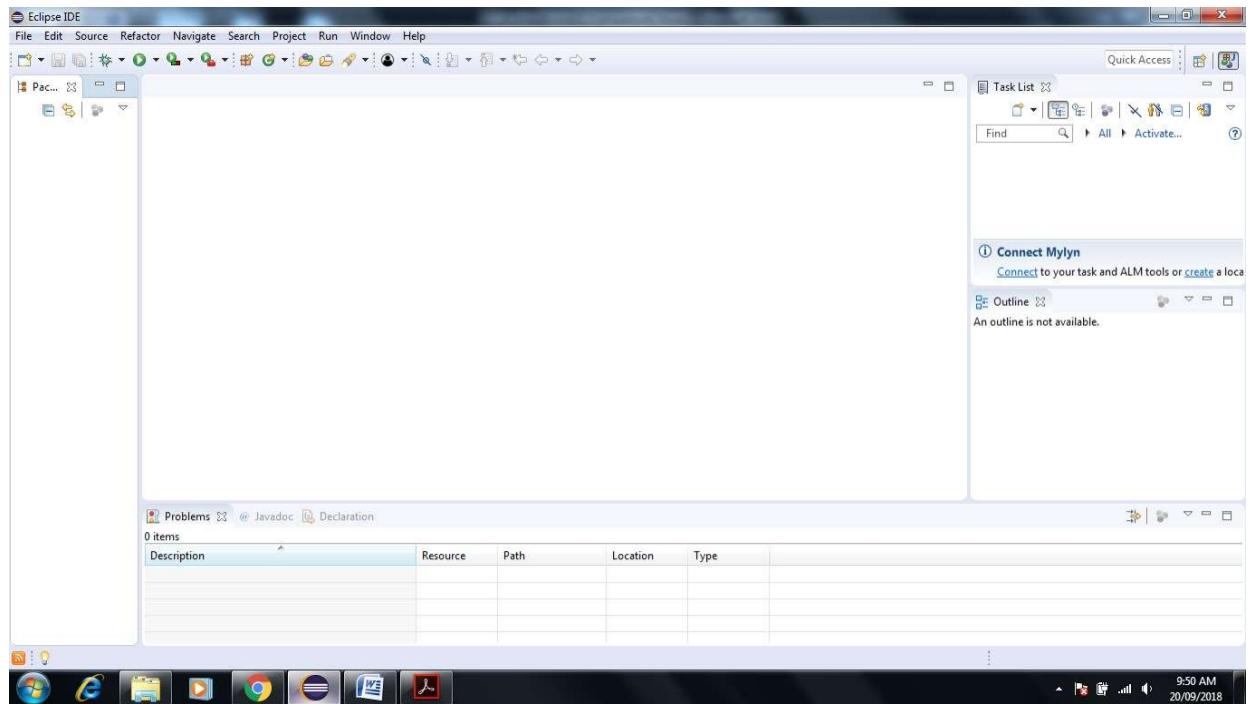
JUnit features include:

- Assertions for testing expected results
- Test fixtures for sharing common test data
- Test suites for easily organizing and running tests
- Graphical and textual test runners

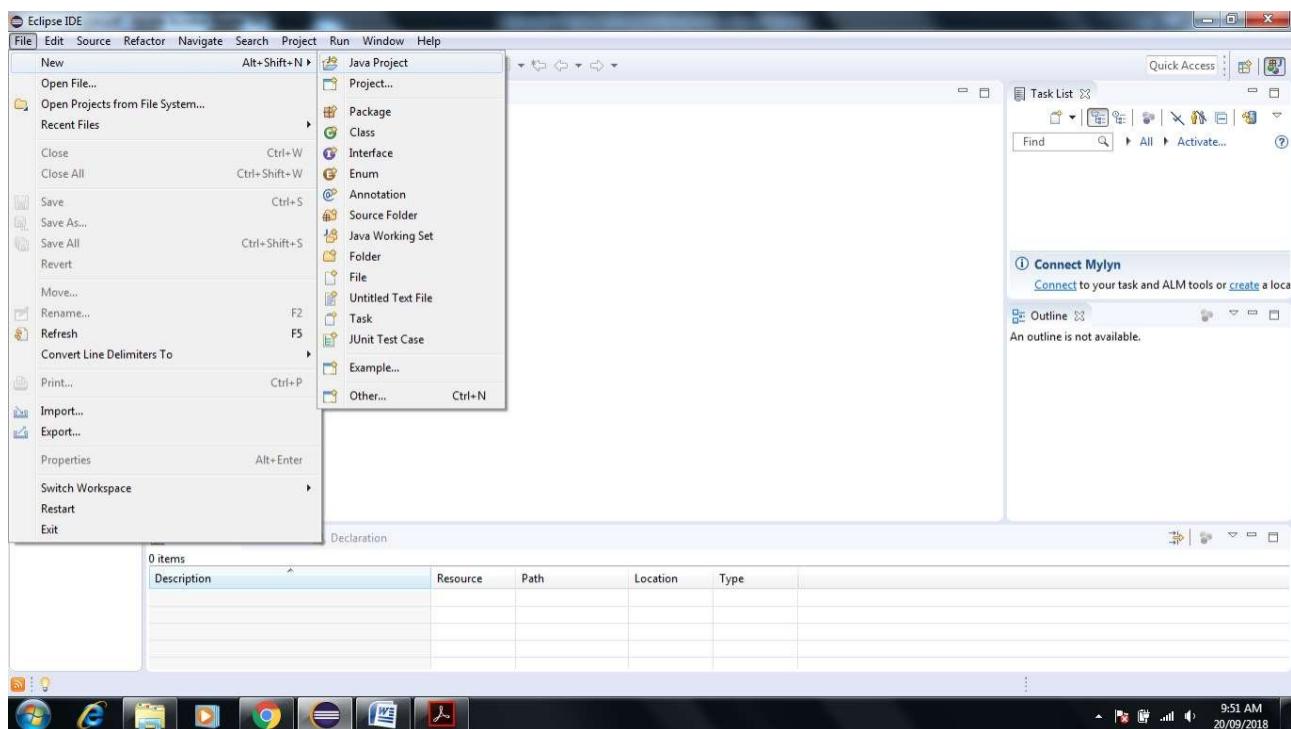
1.8.7 How to Create Simple Junit Test in Eclipse IDE

1. Download JDK 1.8 and Eclipse latest version here we are using eclipse-java-photon-R-win32.

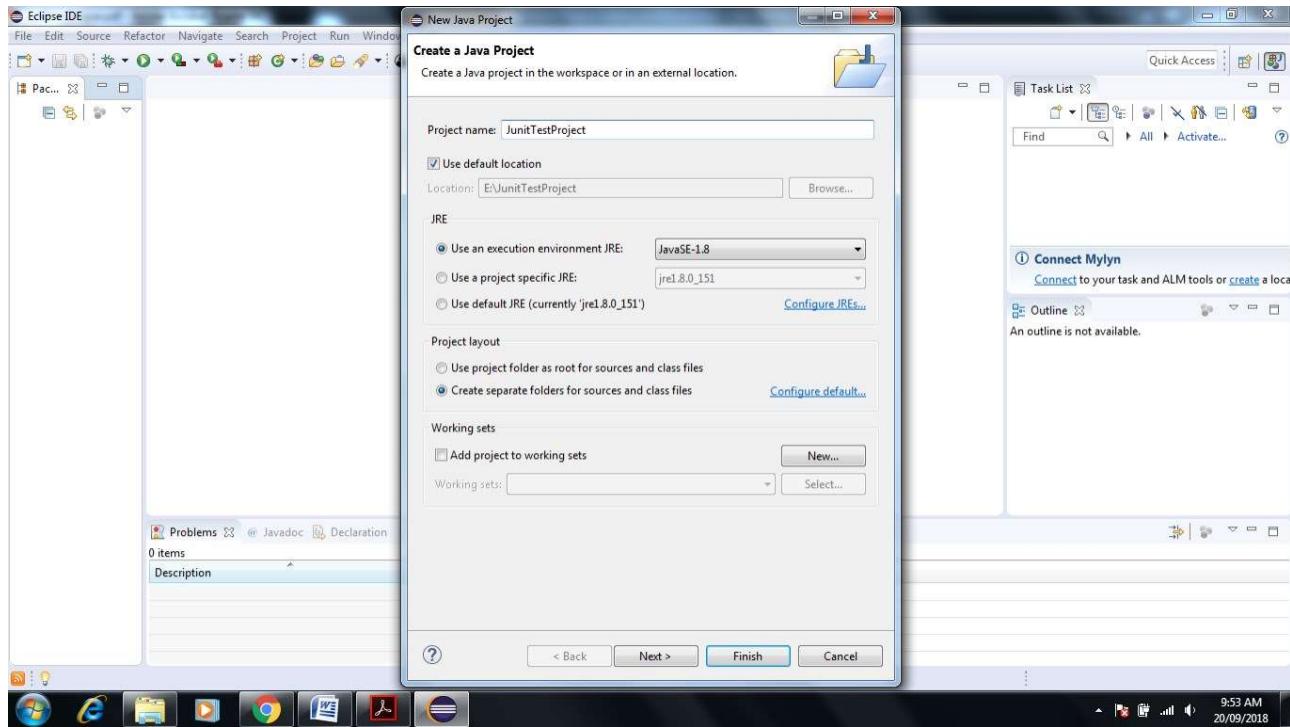
2. Open Eclipse IDE



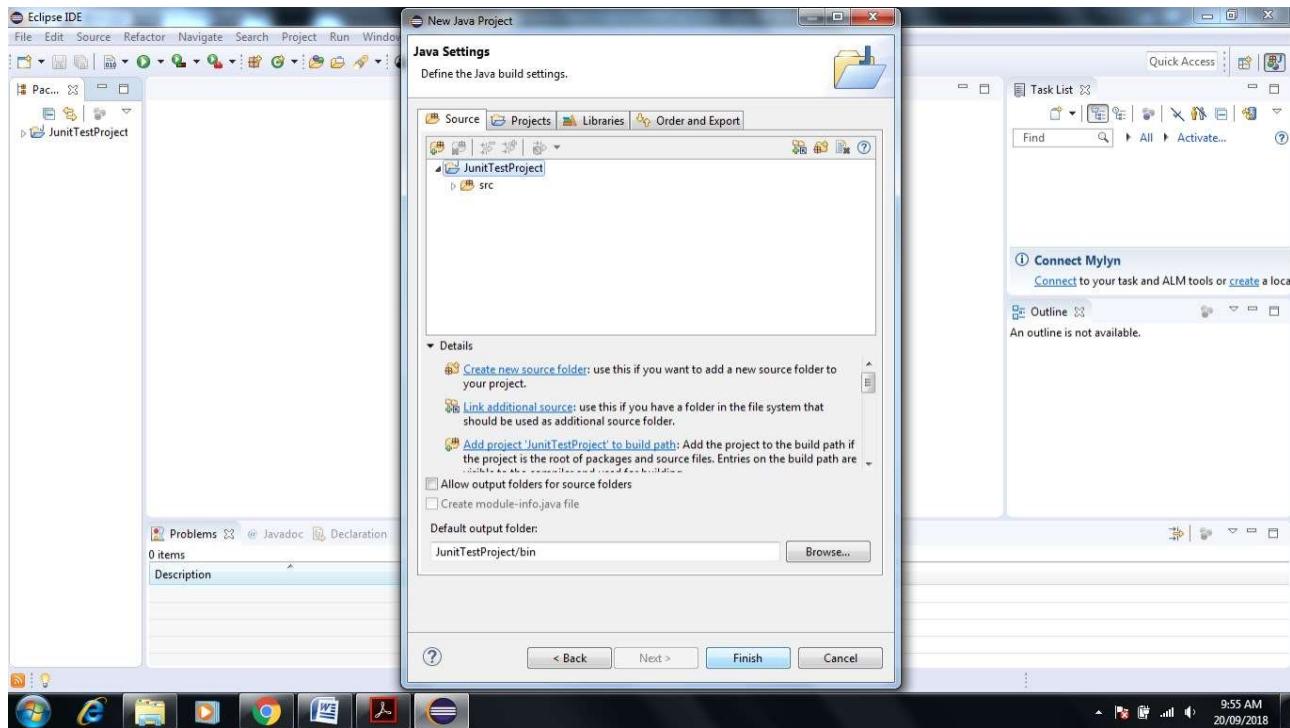
3. Go to File and Select New -> Create New Java Project



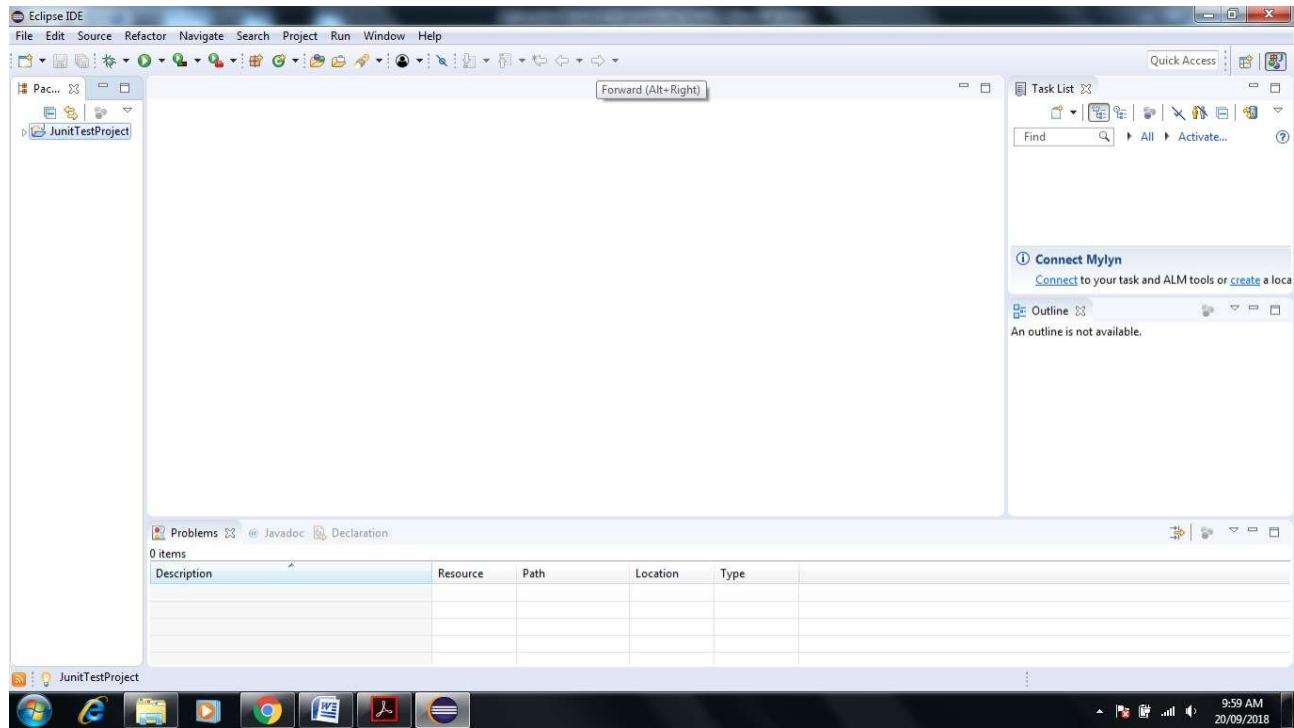
4. Give JunitTestProject name to the project and check use project folder as root for source and class files



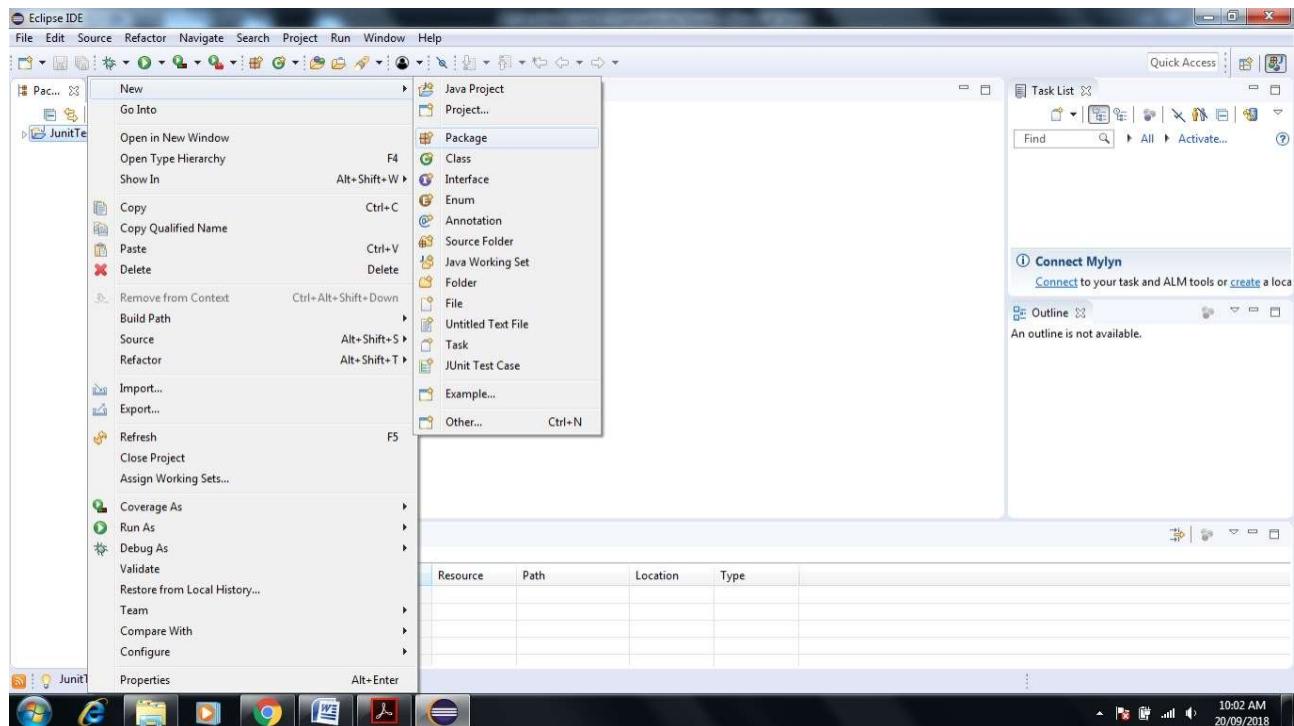
5. Click on Next-> Next Screen will Appear-> Click Finish



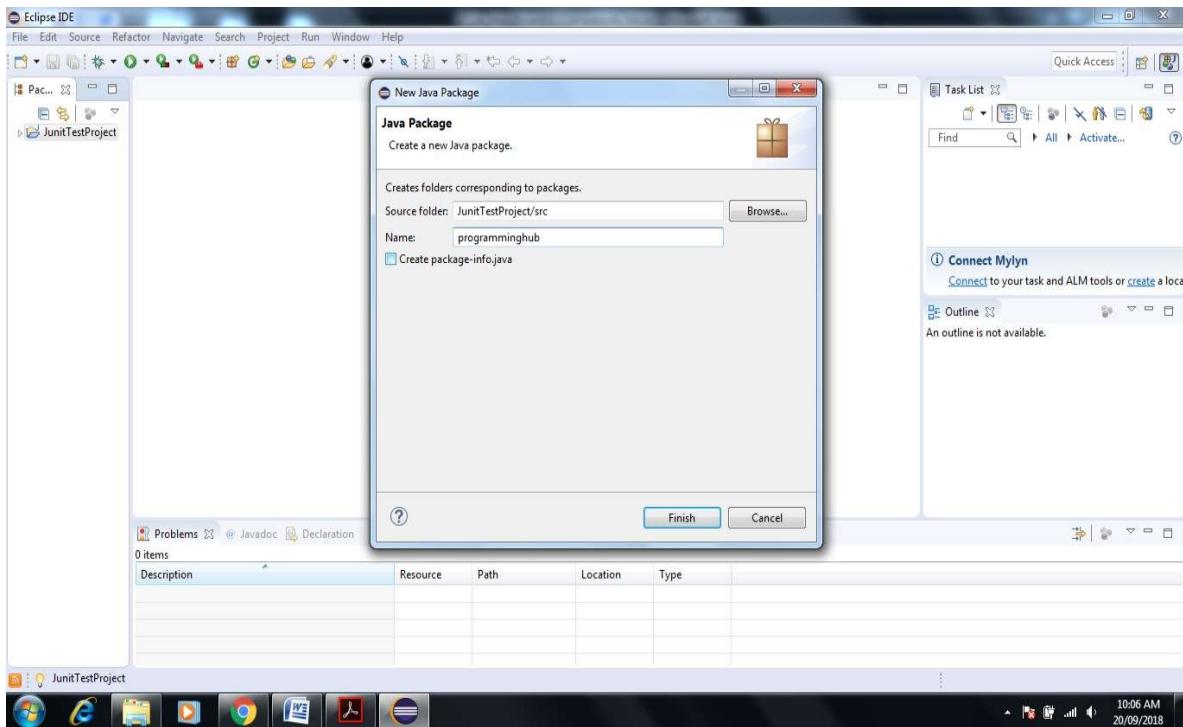
6. Next Screen Shown JunitTestProject Folder in Project Explorer



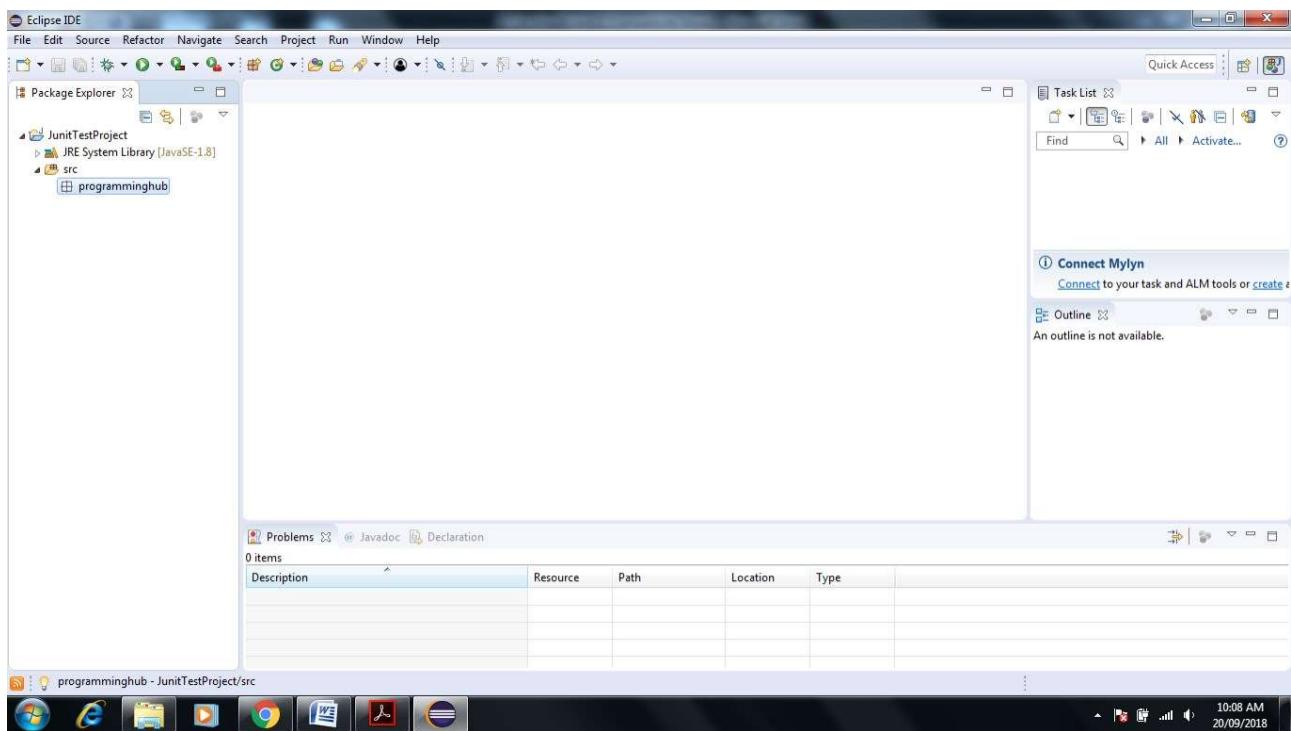
7. Right Click on Folder name JunitTestProject->New->Package



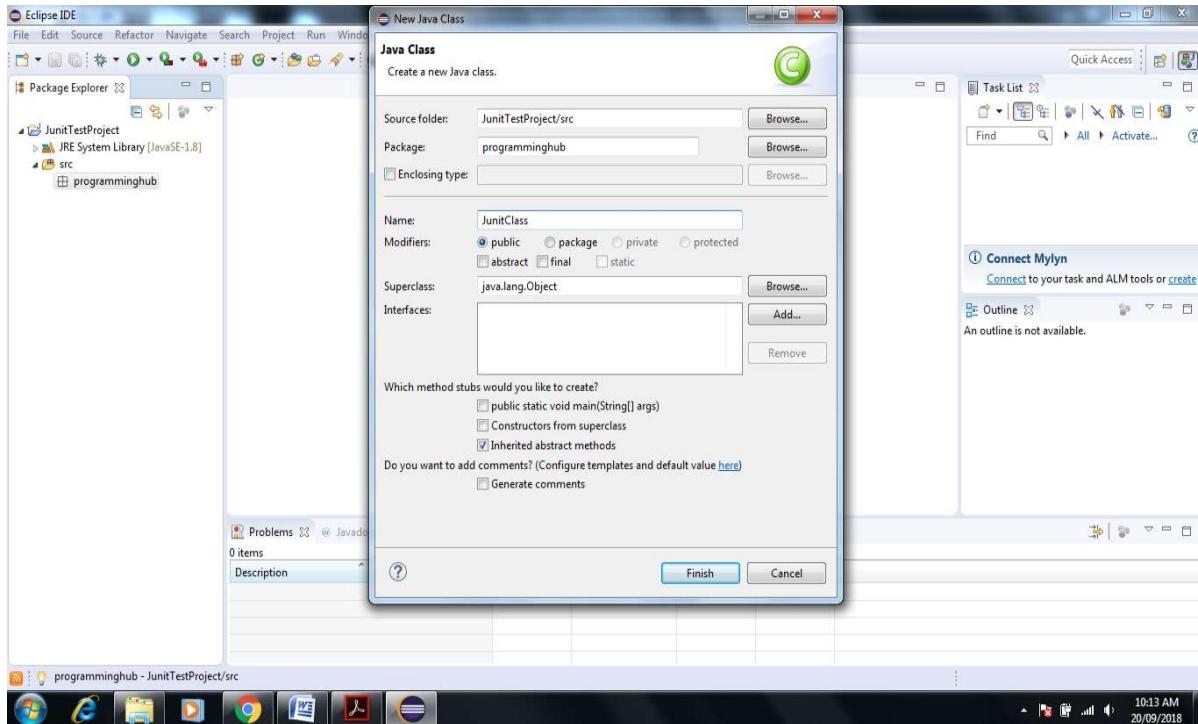
8. Name package as programming hub-> Click on Finish



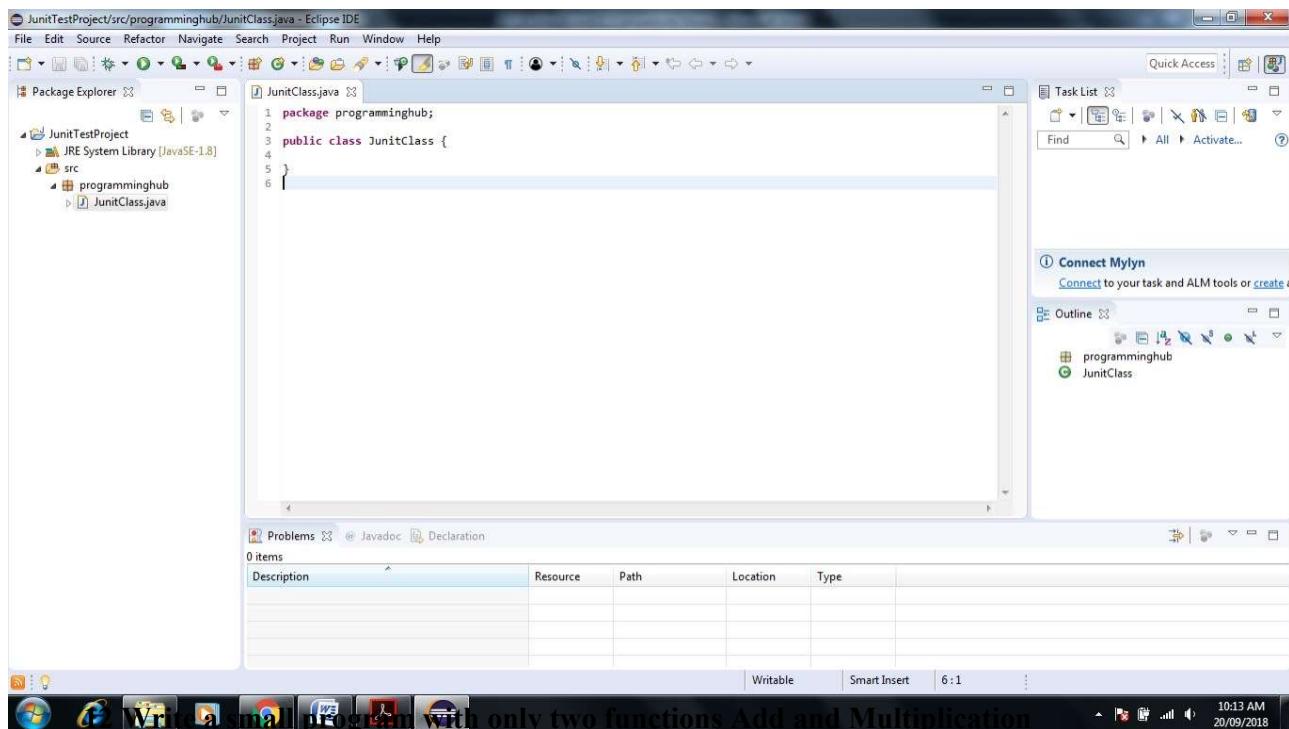
9. See the Programming hub package see in project Explorer Screen of Eclipse



10. Right Click on Programminghub Package->New->Class give the name JunitClass->Click Finish.



11. Next screen will appear



12 Write a small program with only two functions Add and Multiplication

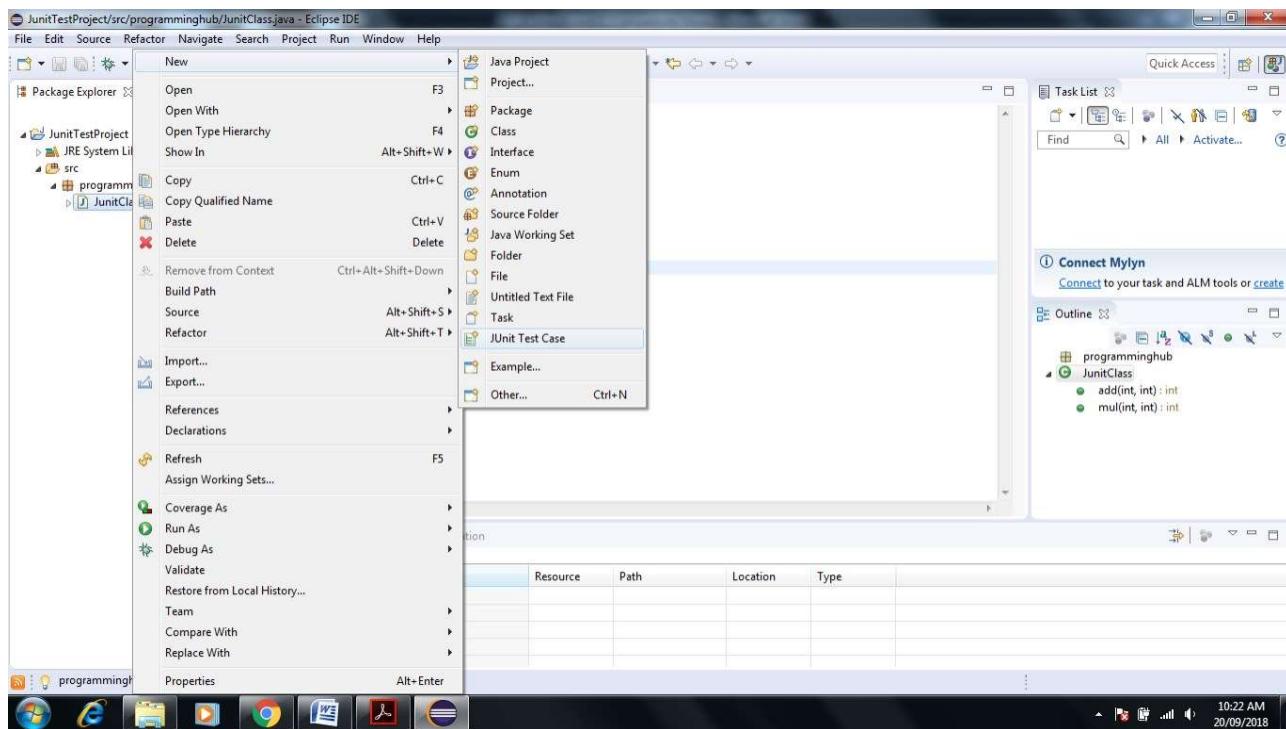
The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The left sidebar has a Package Explorer showing a project named JUnitTestProject with a src folder containing a programminghub package and a JUnitClass.java file. The main editor window displays the following Java code:

```
1 package programminghub;
2
3 public class JUnitClass {
4
5     public int add(int a,int b) {
6         return a+b;
7     }
8     public int mul(int a,int b) {
9         return a*b;
10    }
11 }
```

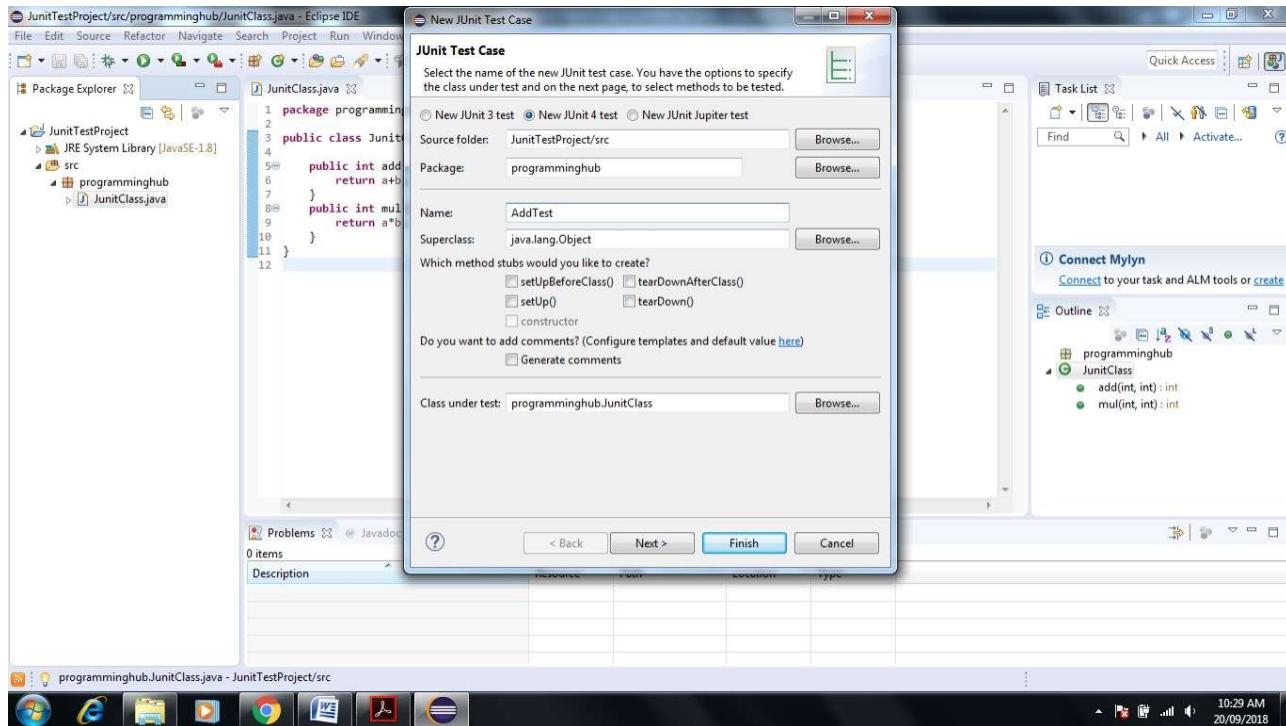
The bottom status bar shows the date and time as 10:19 AM 20/09/2018.

13. Write Test Cases for Java Program

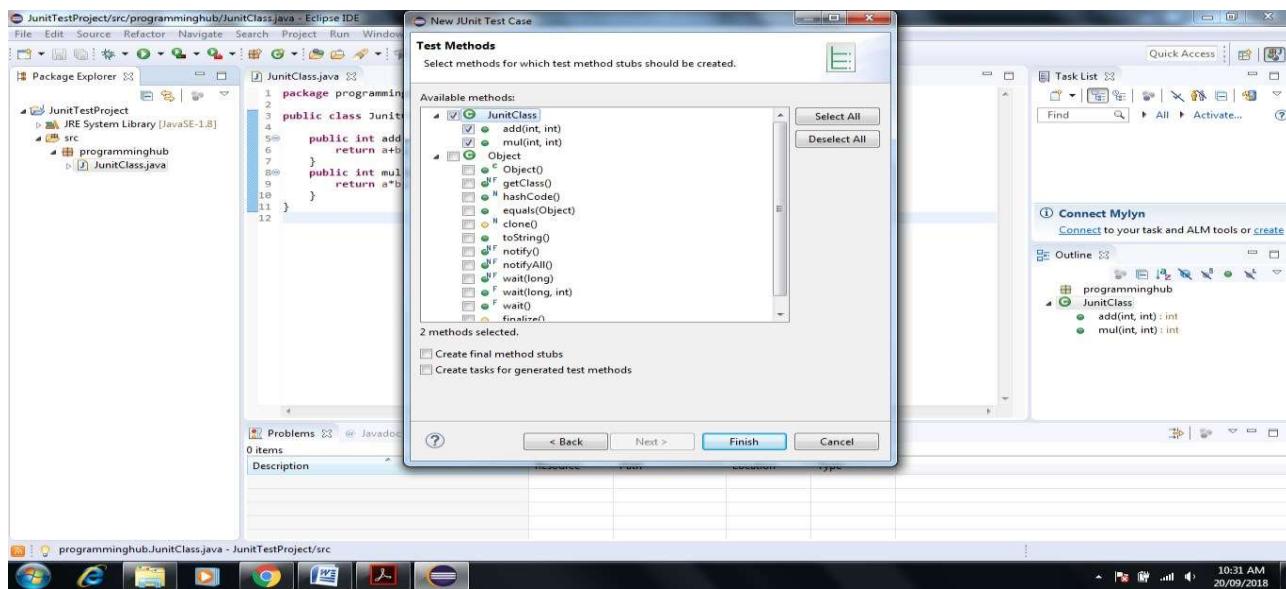
Right click on Junitclass-> New-> Click on Junit Test Cases



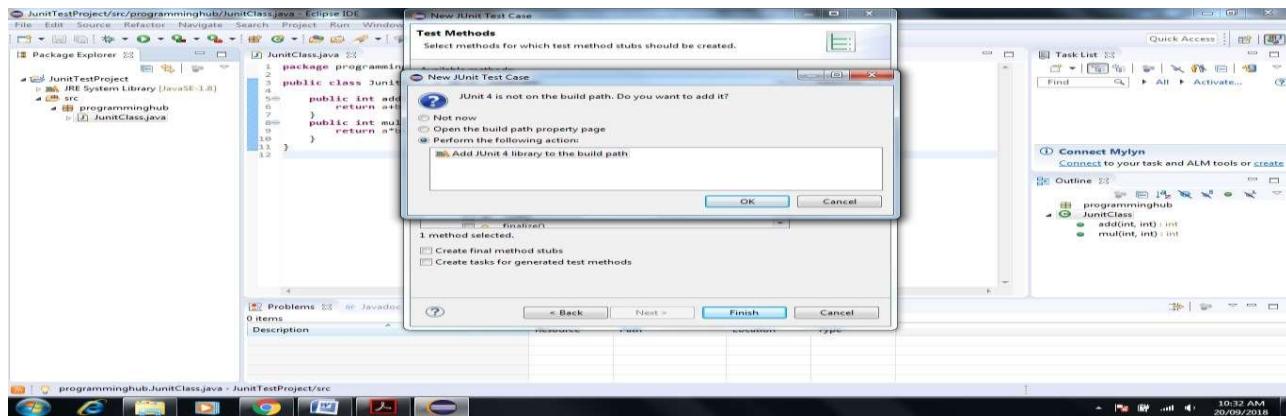
14. Name test suite as AddTest and choose New Junit4 test



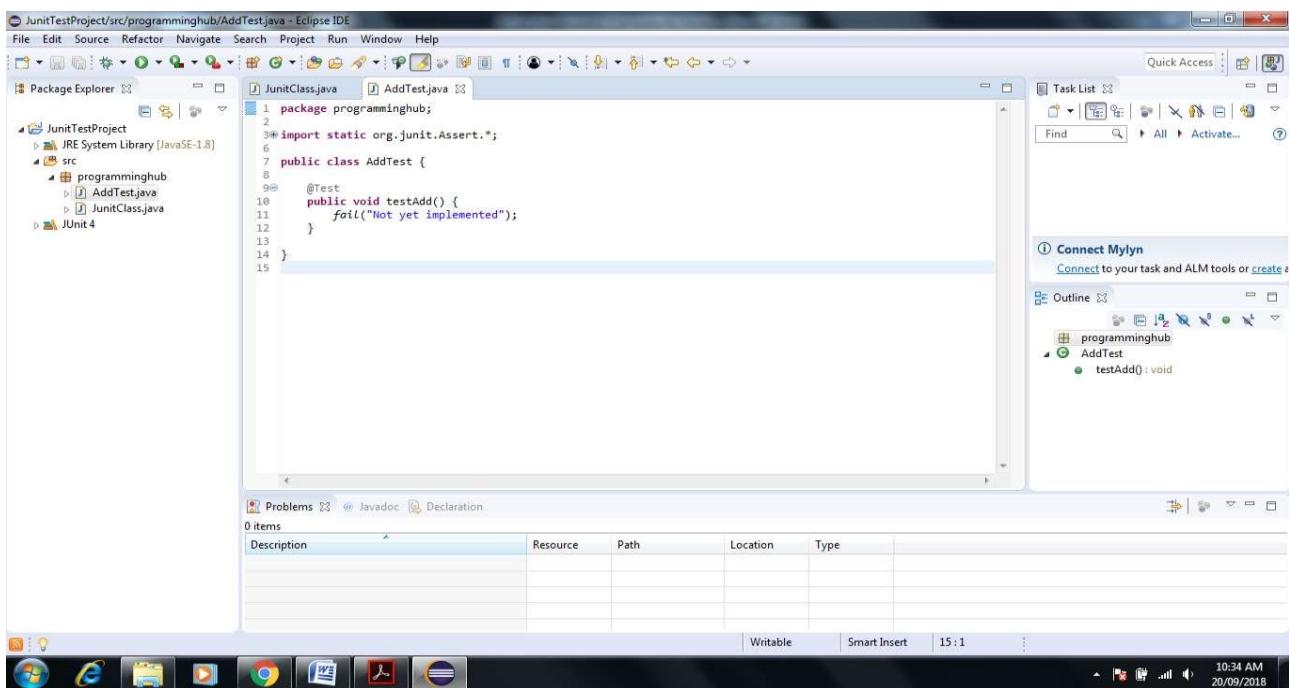
15. Click on add Checkbox



16. Click on Next-> Ok



17. Next screen will appear



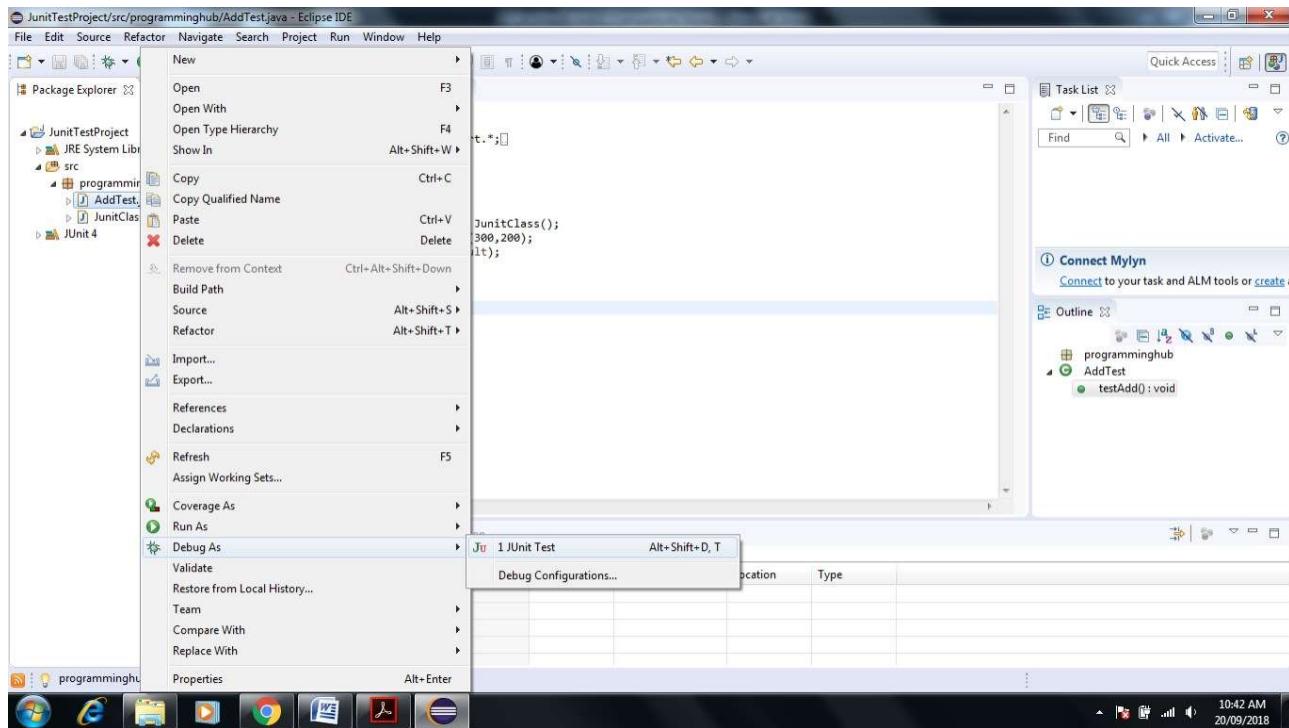
18. Write a code for Test case addition of two number inside AddTest

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** JunitTestProject/src/programminghub/AddTest.java - Eclipse IDE
- Menu Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help
- Toolbar:** Standard Eclipse toolbar icons.
- Package Explorer:** Shows the project structure with JunitTestProject, src, programminghub, AddTest.java, JunitClass.java, and JUnit 4.
- Editor:** Displays the Java code for AddTest.java:

```
1 package programminghub;
2
3 import static org.junit.Assert.*;
4
5 public class AddTest {
6
7     @Test
8     public void testAdd() {
9         JunitClass junit=new JunitClass();
10        int result=junit.add(300,200);
11        assertEquals(500,result);
12    }
13
14 }
15
16 }
```
- Outline View:** Shows the class AddTest and its method testAdd().
- Problems View:** Shows 0 items.
- Task List:** Shows a message to connect Mylyn.
- Bottom Status Bar:** Shows the date and time: 10:39 AM 20/09/2018.

19. Let us run AddTest test case. Right click AddTest-> Debug As->JUnit Test



20. Result of test case is as follows. It shows 0 error and 0 failure and green color test bar which means that test case has run successfully(Green Color Bar Indicate)

The screenshot shows the Eclipse IDE interface with the title "JUnitTestProject/src/programminghub/AddTest.java - Eclipse IDE". The "Package Explorer" view shows two files: "JunitClass.java" and "AddTest.java". The "AddTest.java" code is:

```
1 package programminghub;
2
3 import static org.junit.Assert.*;
4
5 public class AddTest {
6
7     @Test
8     public void testAdd() {
9         JunitClass junit=new JunitClass();
10        int result=junit.add(300,200);
11        assertEquals(500,result);
12    }
13
14 }
15
16
17 }
```

The "Run" view in the top left shows "Runs: 1/1", "Errors: 0", and "Failures: 0", indicating a successful run. The "Problems" view shows 0 items. The "Outline" view on the right shows the class structure. The status bar at the bottom right shows the date and time as "20/09/2018 10:44 AM".

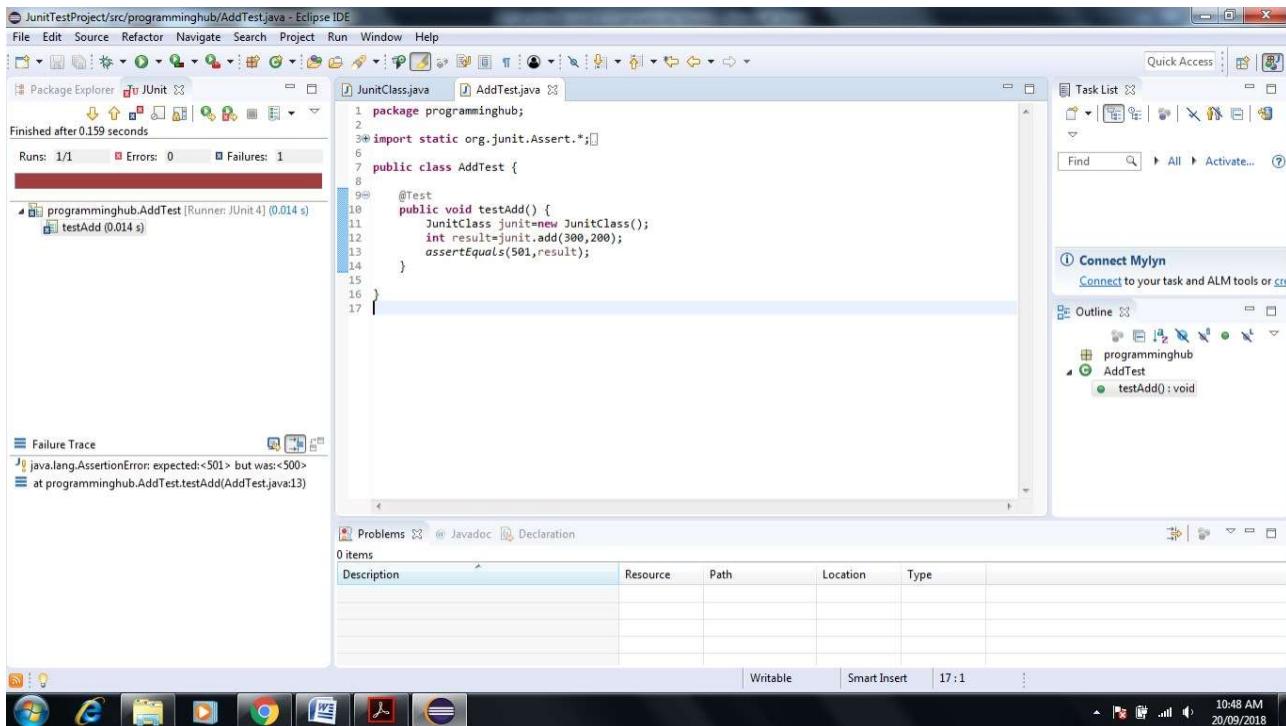
21. Let us purposely give wrong input in assertEquals method or unexpected result here we write 501 instead of 500 indicate wrong addition result

The screenshot shows the Eclipse IDE interface with the title "JUnitTestProject/src/programminghub/AddTest.java - Eclipse IDE". The "Package Explorer" view shows two files: "JunitClass.java" and "AddTest.java". The "AddTest.java" code is identical to the previous screenshot:

```
1 package programminghub;
2
3 import static org.junit.Assert.*;
4
5 public class AddTest {
6
7     @Test
8     public void testAdd() {
9         JunitClass junit=new JunitClass();
10        int result=junit.add(300,200);
11        assertEquals(501,result);
12    }
13
14 }
15
16
17 }
```

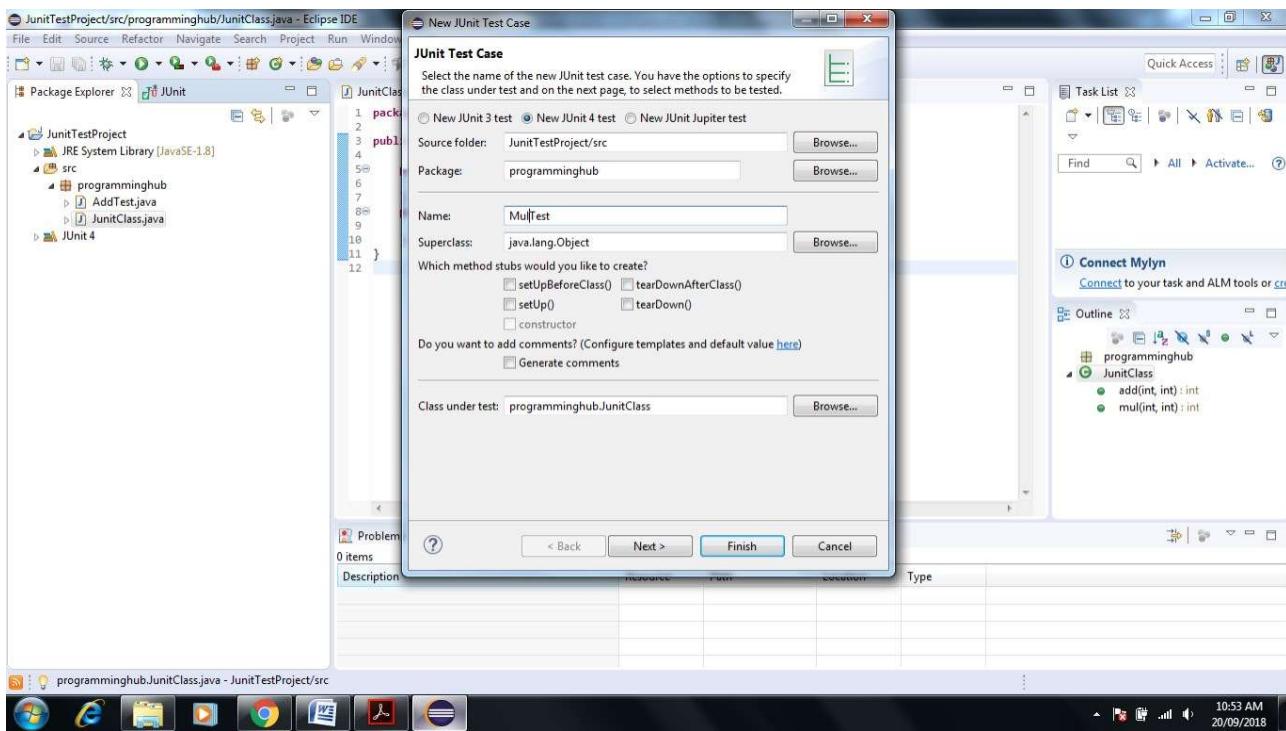
The "Run" view in the top left shows "Runs: 1/1", "Errors: 0", and "Failures: 0", indicating a successful run. However, the "Problems" view shows 1 item with the message "java.util.AssertionError: expected [501] but found [500]", indicating a failure. The "Outline" view on the right shows the class structure. The status bar at the bottom right shows the date and time as "20/09/2018 10:46 AM".

22. Now test case should fail.(Brown Color Bar Indicate) So again run AddTest as follows

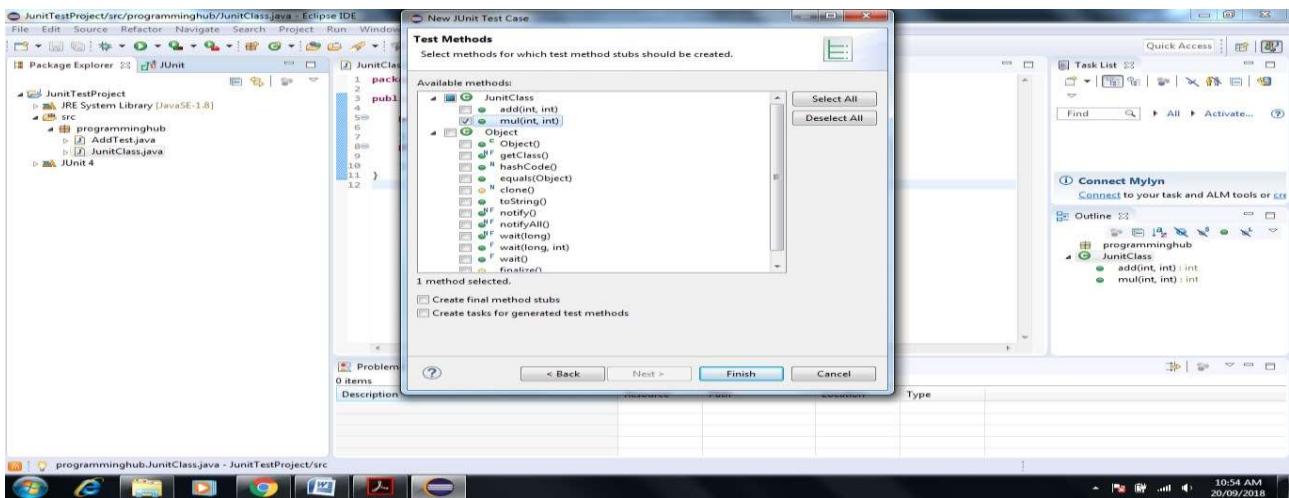


23. Similarly you can Create Test case for Multiplication Function

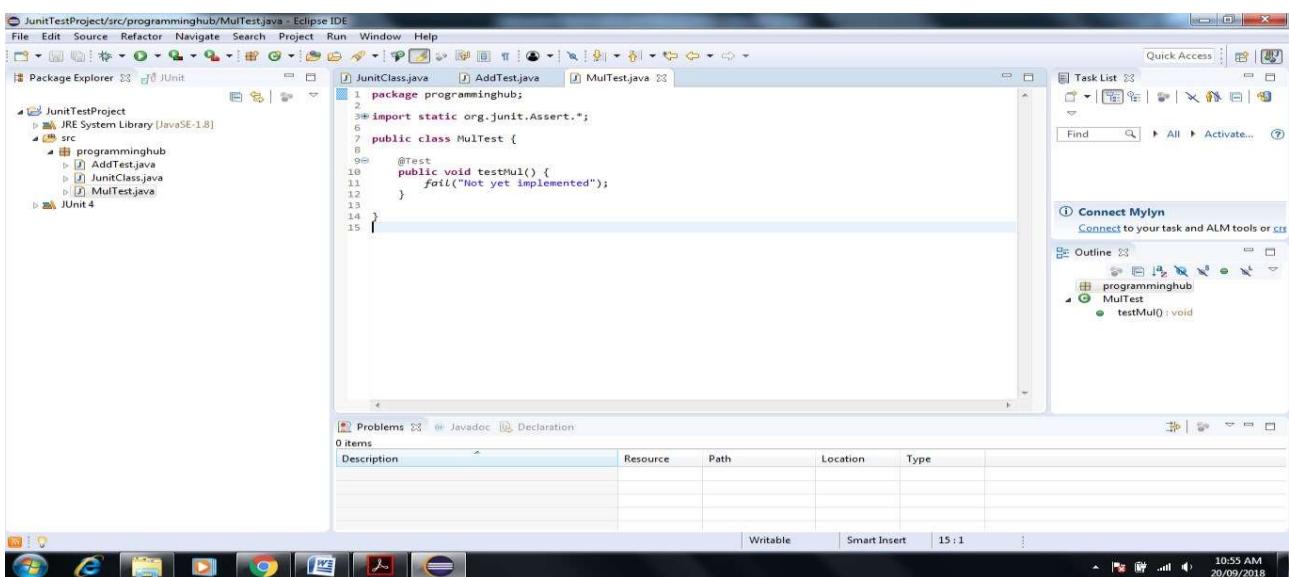
Click on Project Explorer Screen-> Right Click on JunitClass->New->JUnit Test Case-> Give name MulTest.



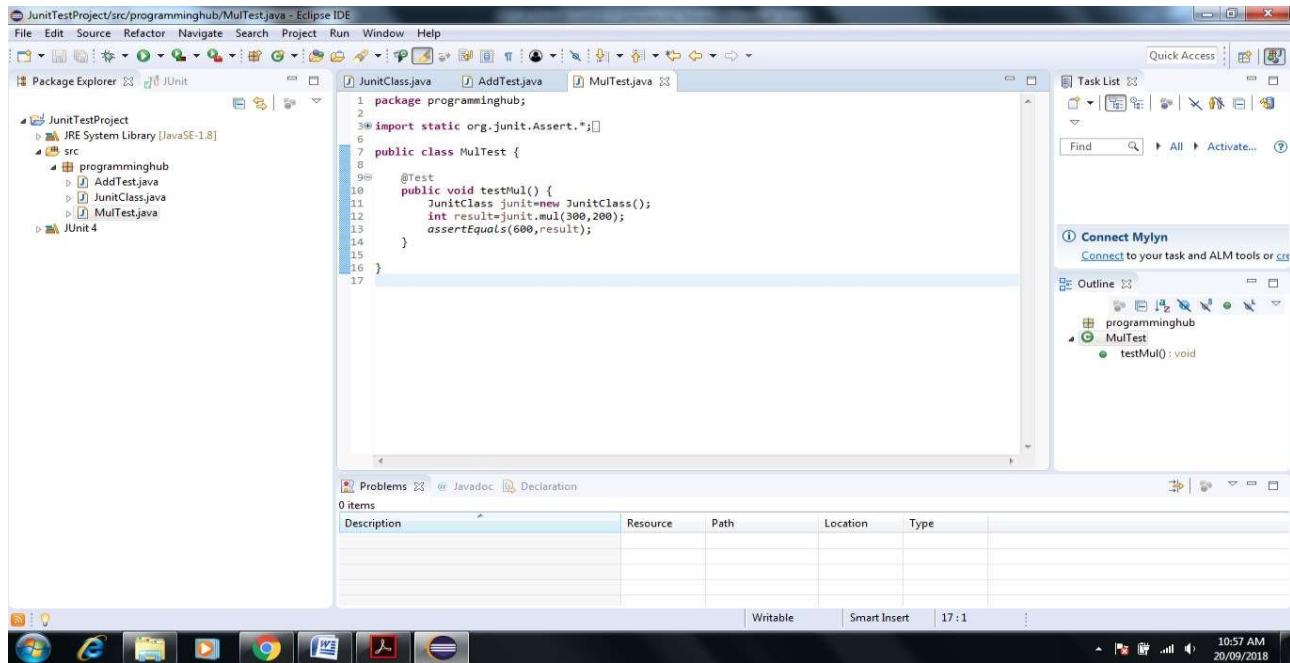
24. Click on Next ->Select Mul Check Box -> Click Finish



25. Next Screen will appear



26. Write a Test Case Code inside MulTest method

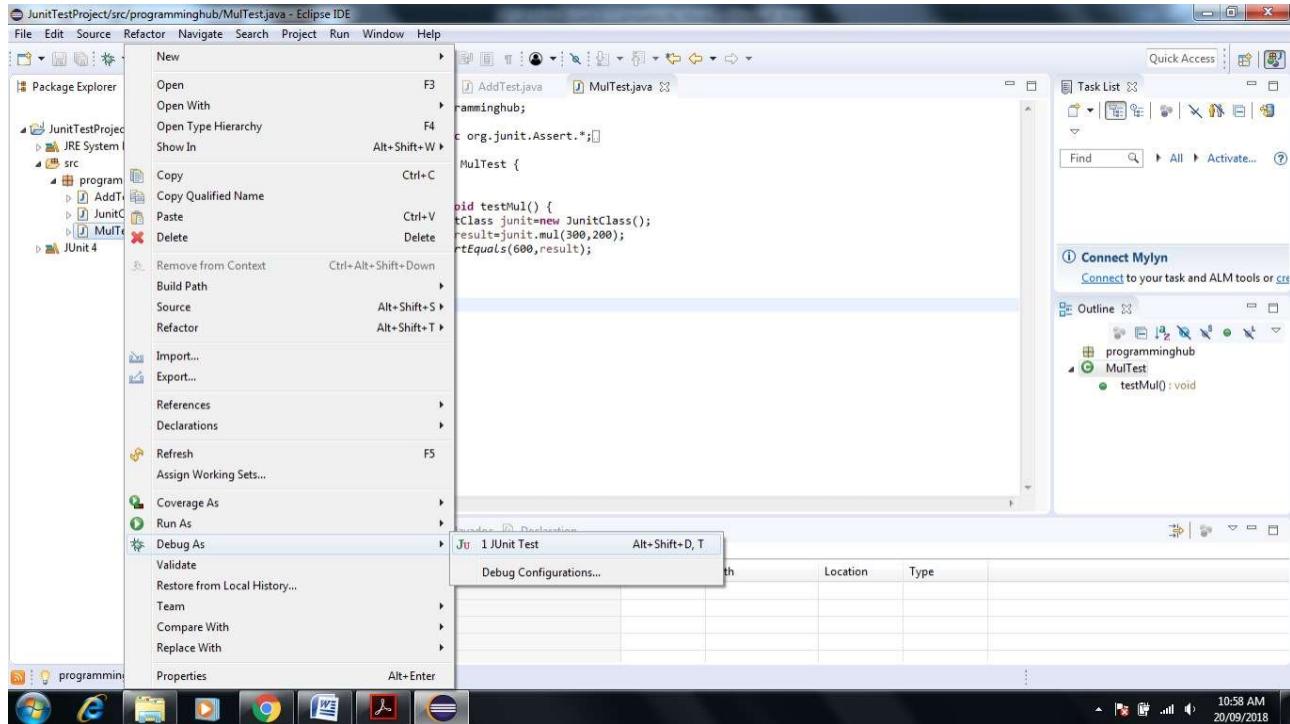


The screenshot shows the Eclipse IDE interface with the following details:

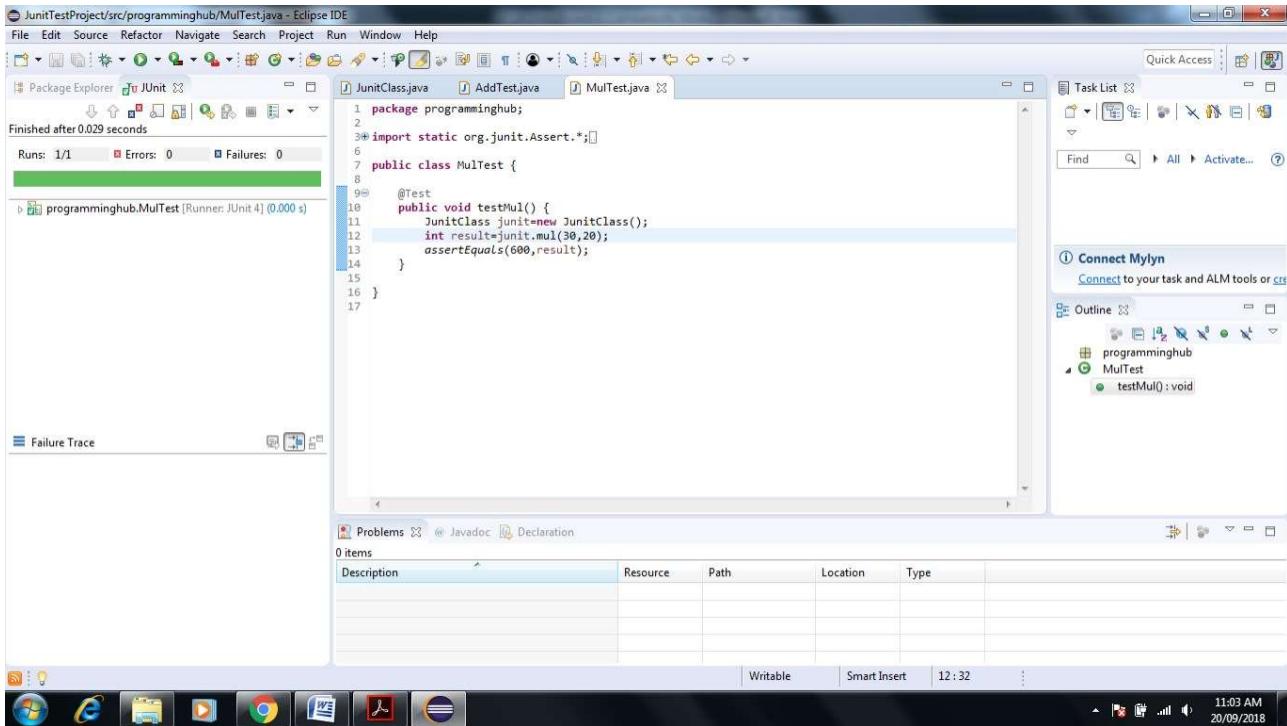
- Project Explorer:** Shows the project "JunitTestProject" with packages "src" and "JUnit". Inside "src", there are files "AddTest.java", "JunitClass.java", and "MulTest.java".
- MulTest.java Content:**

```
1 package programminghub;
2
3 import static org.junit.Assert.*;
4
5 public class MulTest {
6
7     @Test
8     public void testMul() {
9         JunitClass junit=new JunitClass();
10        int result=junit.mul(300,200);
11        assertEquals(600,result);
12    }
13
14 }
15
16 }
```
- Outline View:** Shows the class structure: "programminghub" > "MulTest" > "testMul()".
- Problems View:** Shows 0 items.
- Task List:** Shows a message to "Connect Mylyn".
- Bottom Bar:** Shows the Windows taskbar with various icons and the system clock indicating 10:57 AM on 20/09/2018.

27. Right Click on MulTest->Debug->JUnit Test

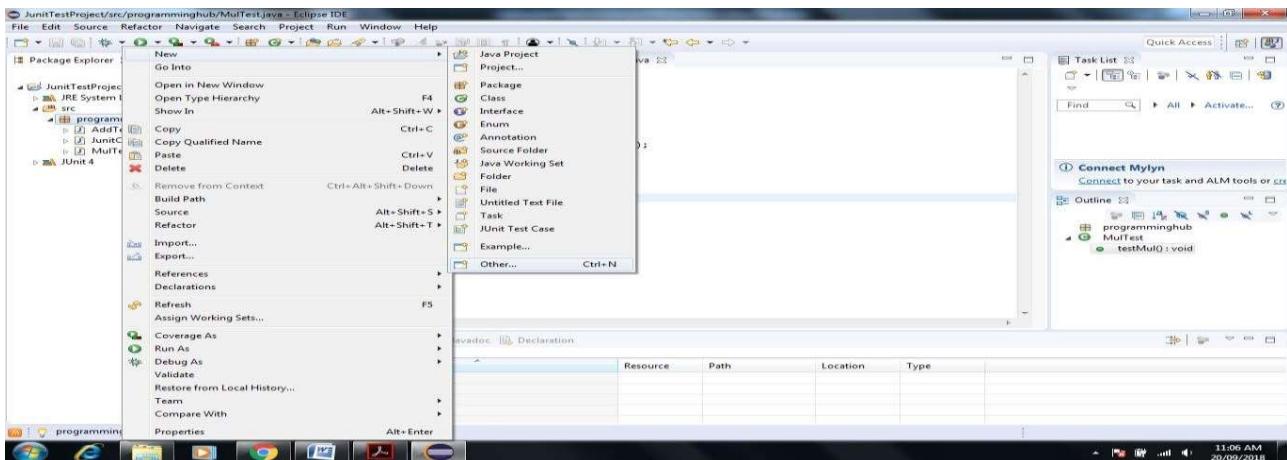


28. Execute Test

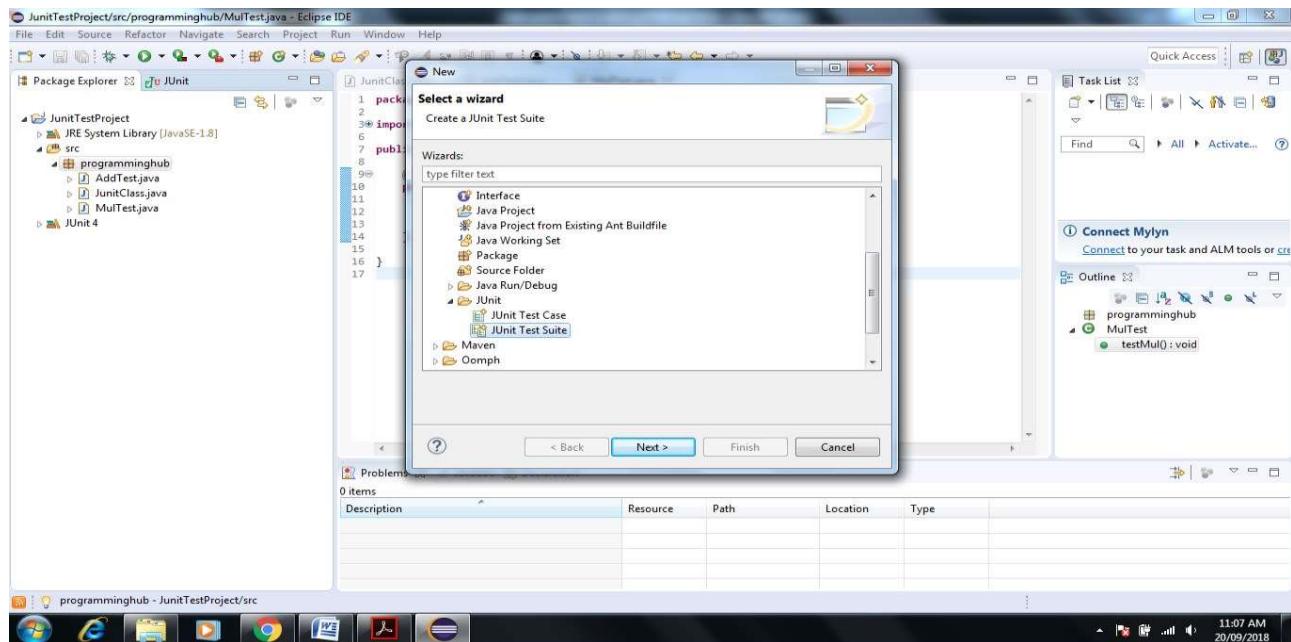


Test Suite – it is used to test multiple test cases at one time.

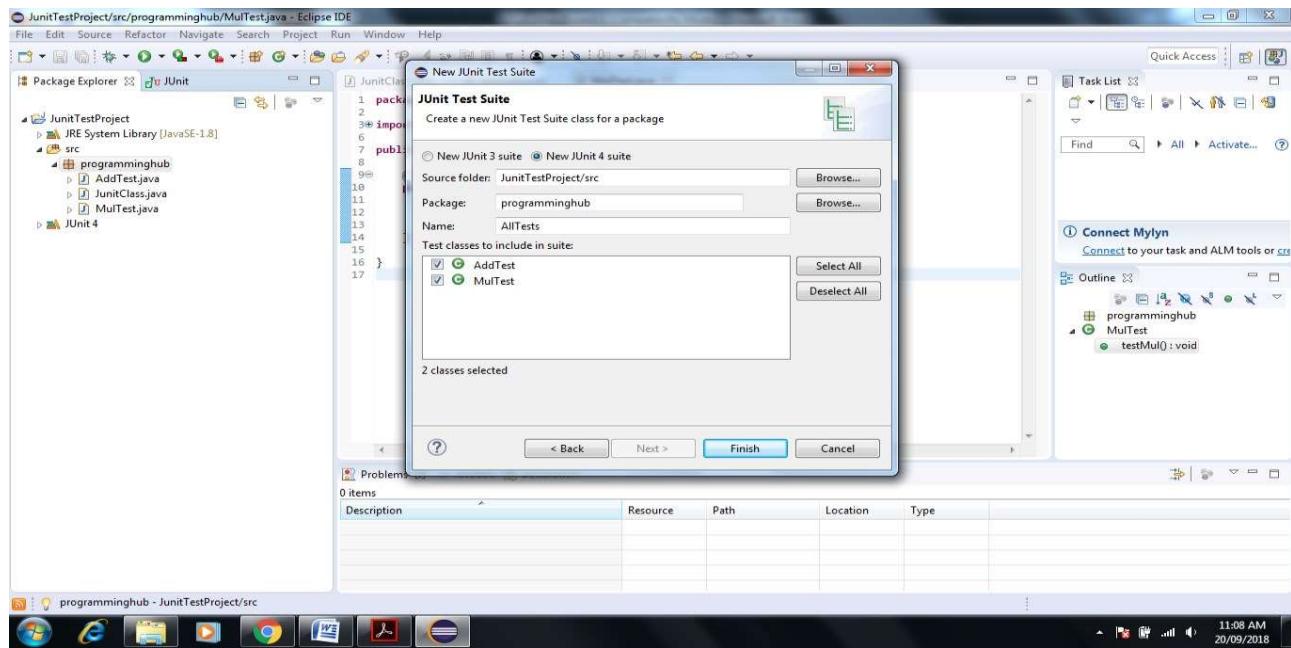
29. Now let us create Test Suite both add and mul test cases in one time



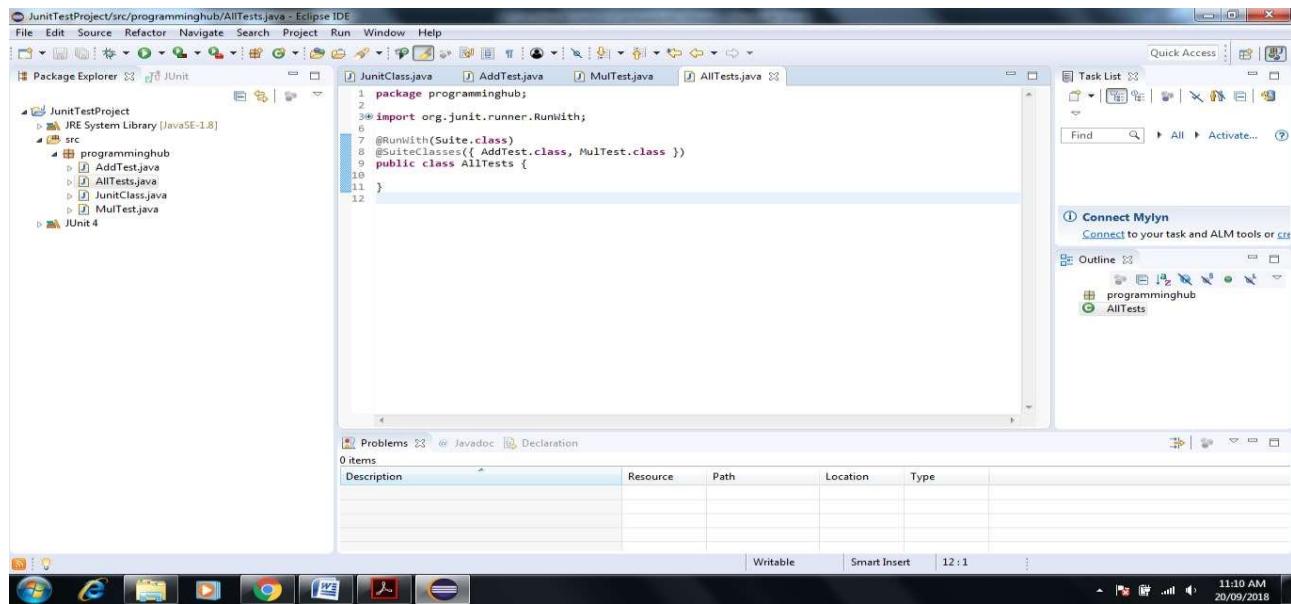
30. Click on Package name->New->Other->JUnit->JUnit Test Suite->Next



31. Click on Finish



32. Next Screen Appear that automatically create Test Suite for Add and Mul

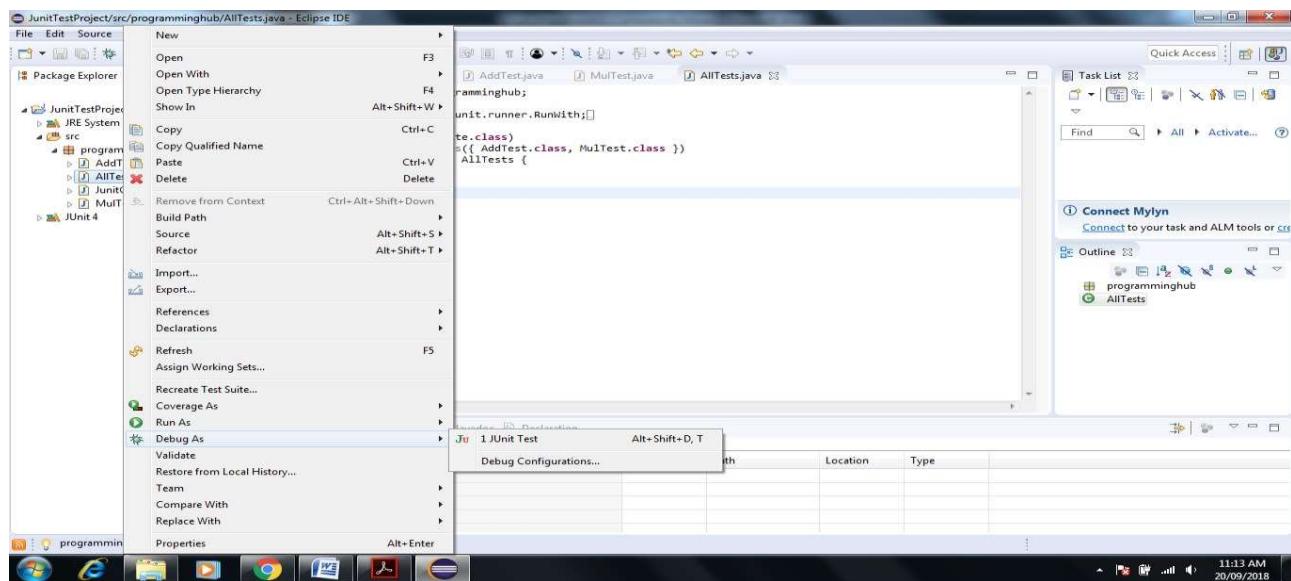


The screenshot shows the Eclipse IDE interface with the title bar "JUnitTestProject/src/programminghub/AllTests.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. The toolbar has icons for New, Open, Save, Cut, Copy, Paste, Find, Replace, and others. The Package Explorer view shows a project named "JUnitTestProject" with packages "src" and "programminghub" containing files "AddTest.java", "AllTests.java", "JunitClass.java", and "MulTest.java". The JUnit view is also present. The central editor window displays the following Java code:

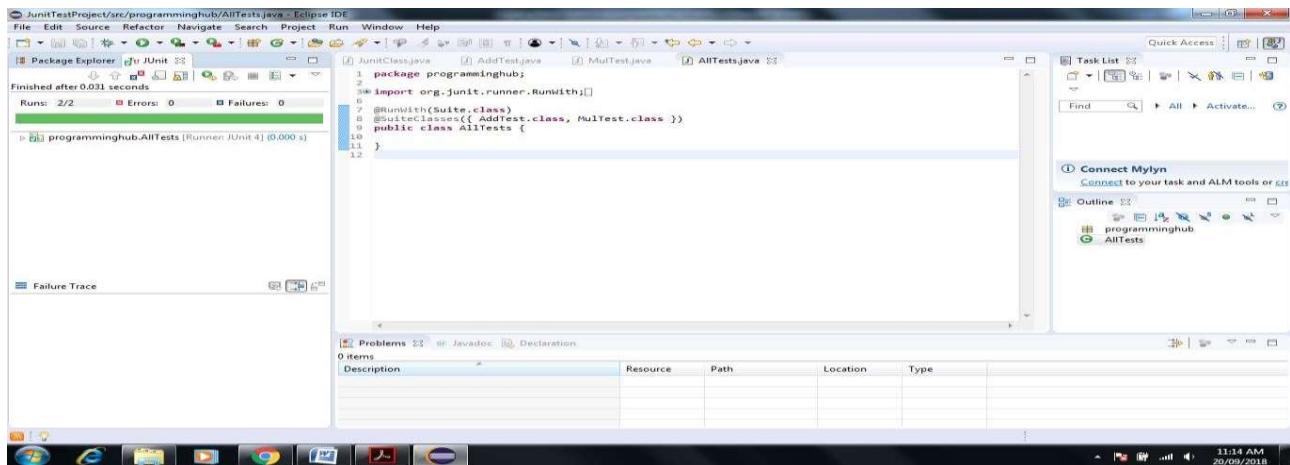
```
1 package programminghub;
2
3 import org.junit.runner.RunWith;
4
5 @RunWith(Suite.class)
6 @SuiteClasses({ AddTest.class, MulTest.class })
7 public class AllTests {
8
9 }
```

The Problems view shows 0 items. The bottom status bar indicates the date and time as 20/09/2018 11:10 AM.

33. Execute Test Suite Right Click on All Test ->Debug->JUnit Test



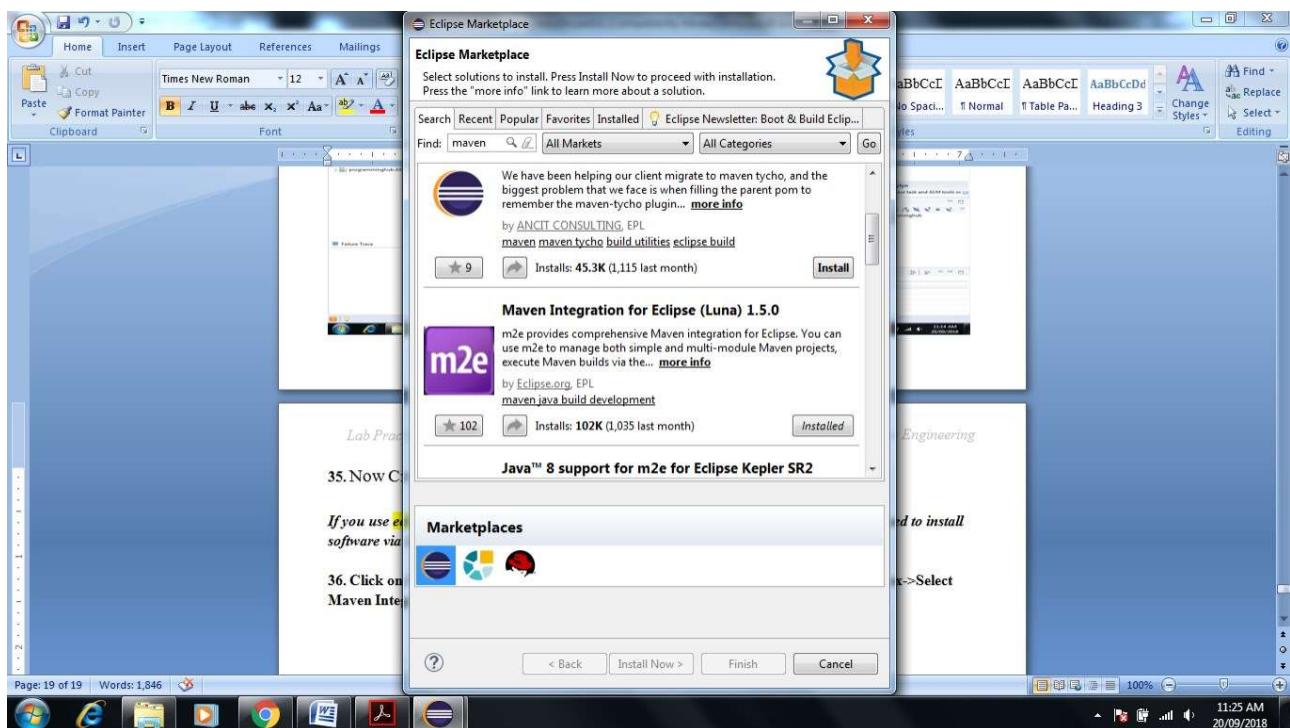
34. Test Suite Executed successfully Test suite fails even if a single test case among all fails.



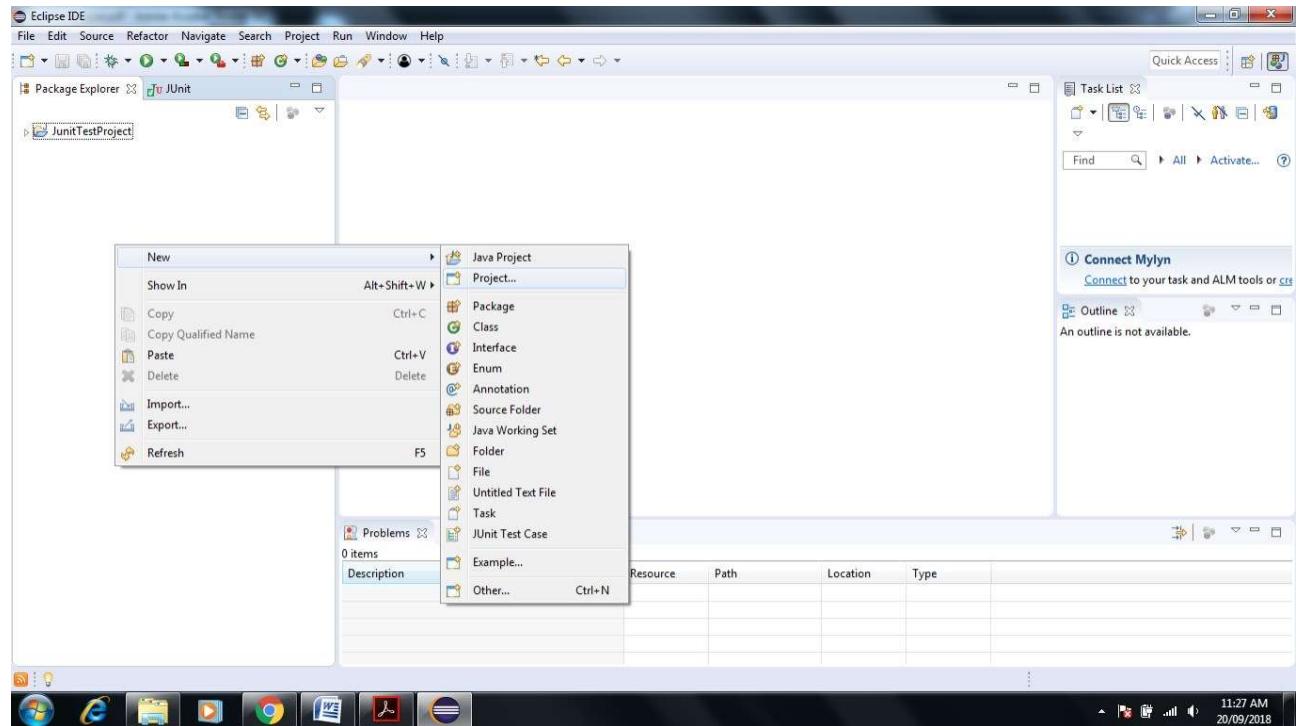
35. Now Create Test Report Using Apache Maven

If you use eclipse-java-photon-R-win32 Version it include Maven in built installed so no need to install software via Eclipse help Install Software Option

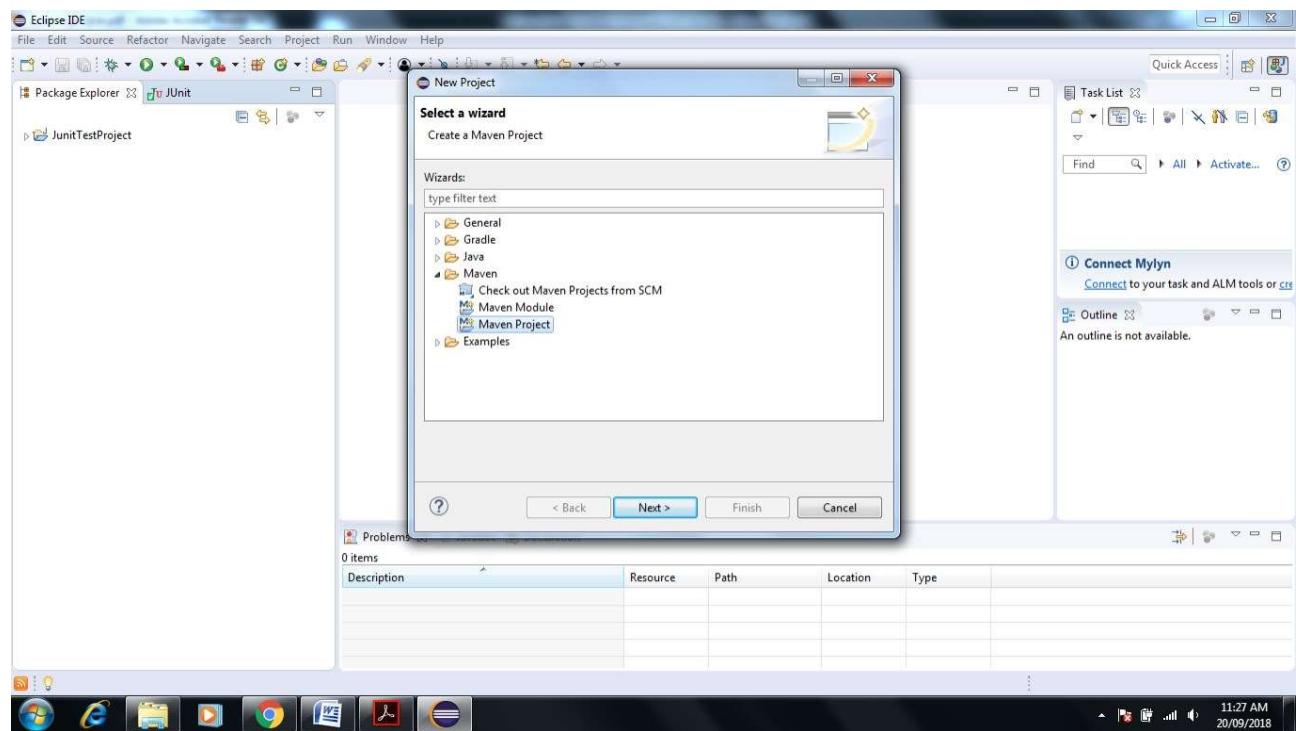
36. Click on Help in Eclipse->Eclipse Marketplace->Enter Maven Keyword in Search box->Select Maven Integration version as per requirement->Click on Install



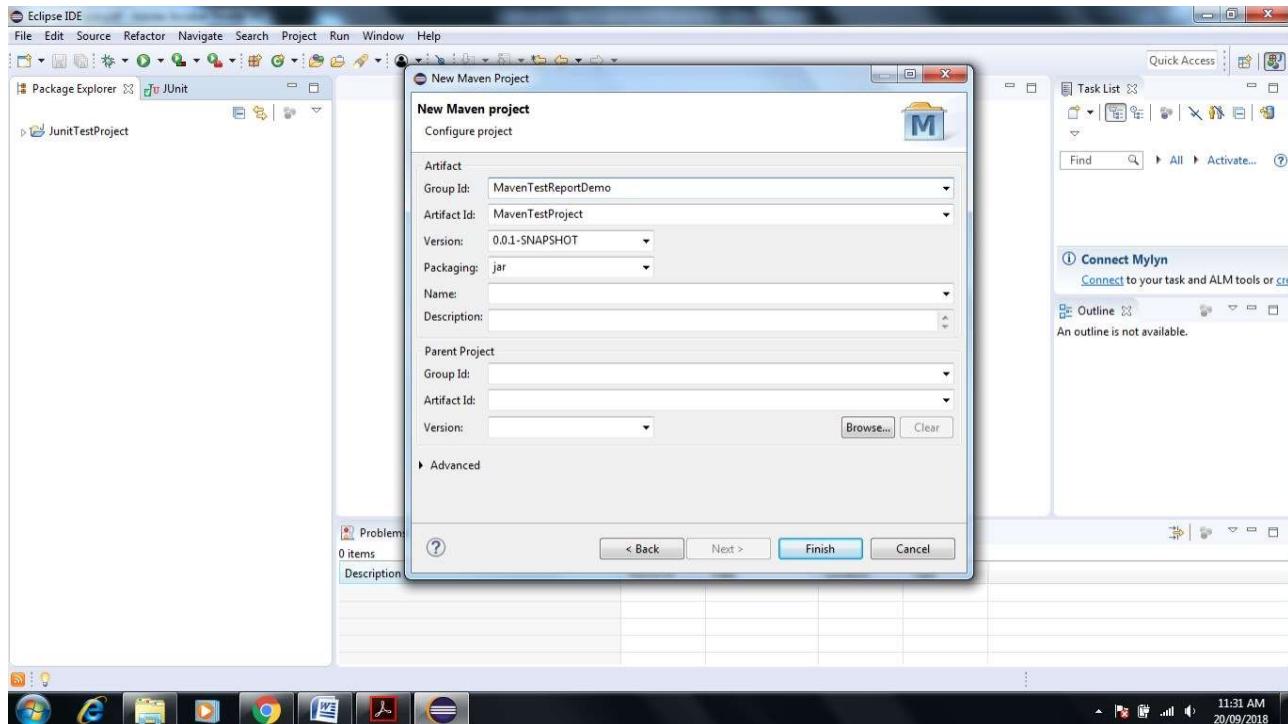
37. Right Click in Project Explorer Window



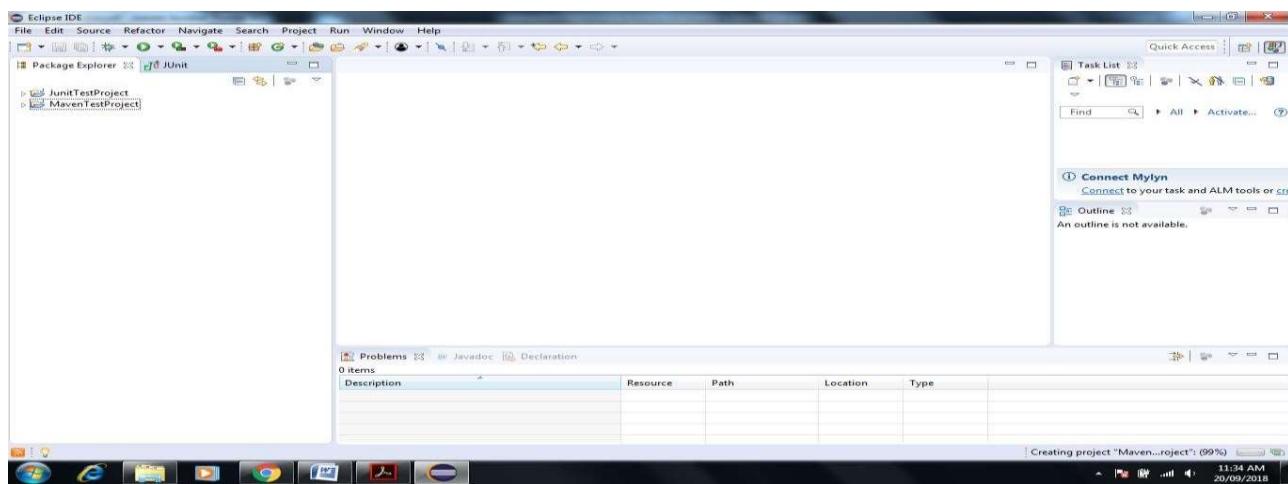
38. Go to Maven Project-> Click Next



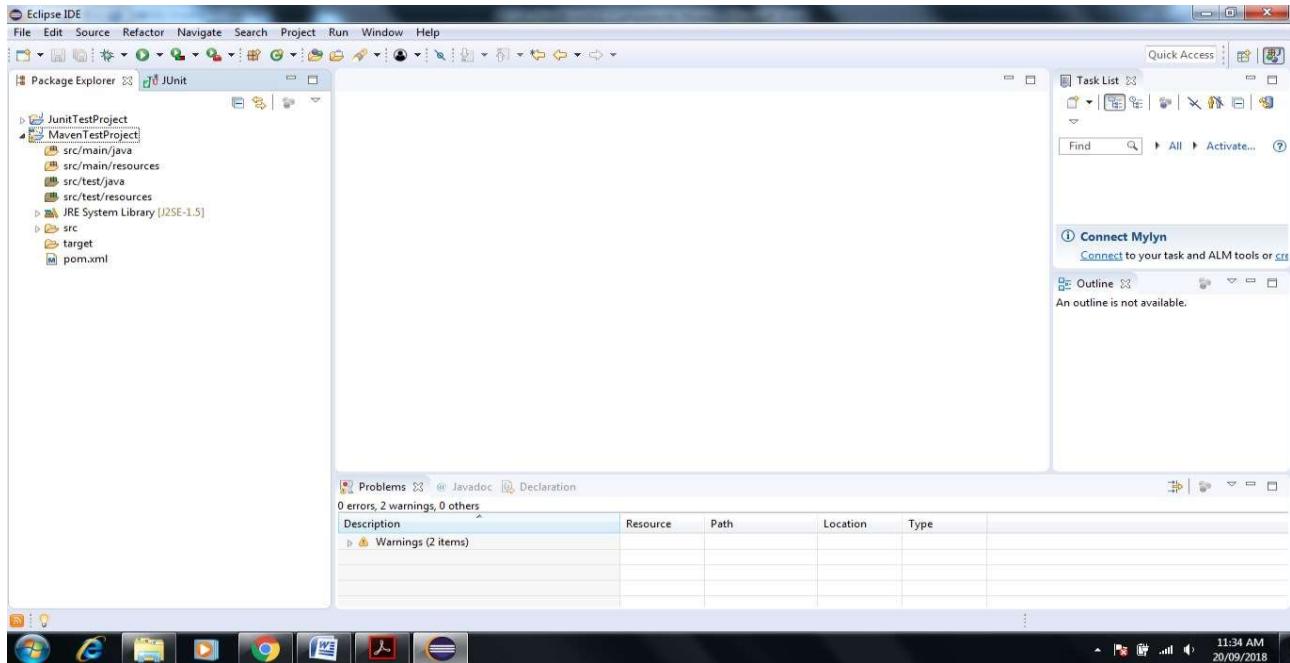
39. Select Check Box Create Simple Project-> Click Next-> Give Group Id and Artifact name



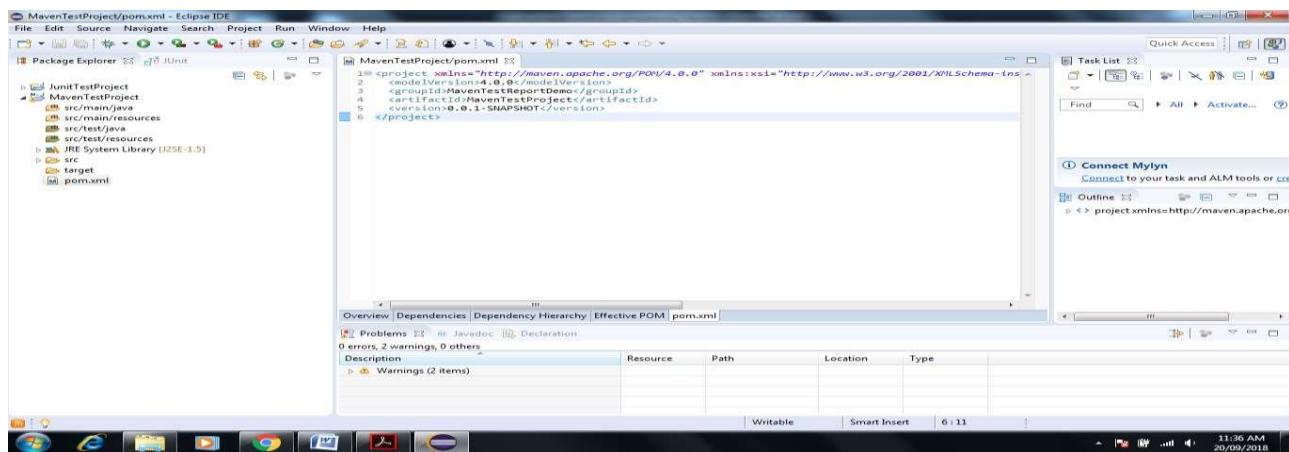
40. Click on Finish-> Next Screen Appear



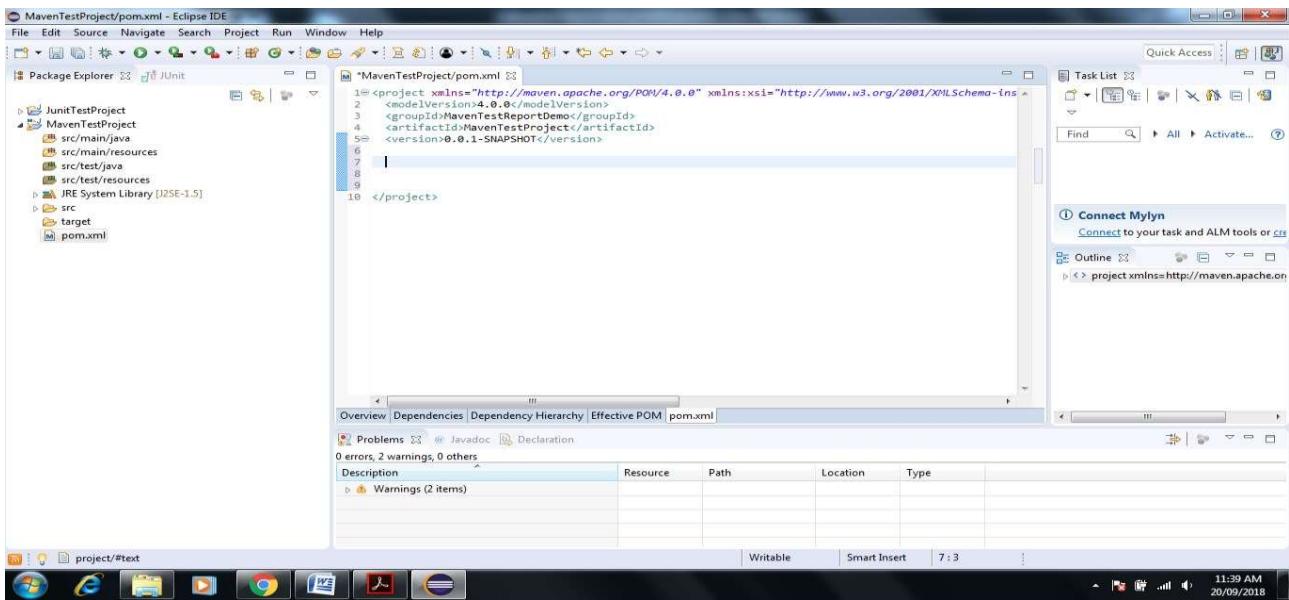
41. MavenTestProject shown Pom.xml file doble click on same



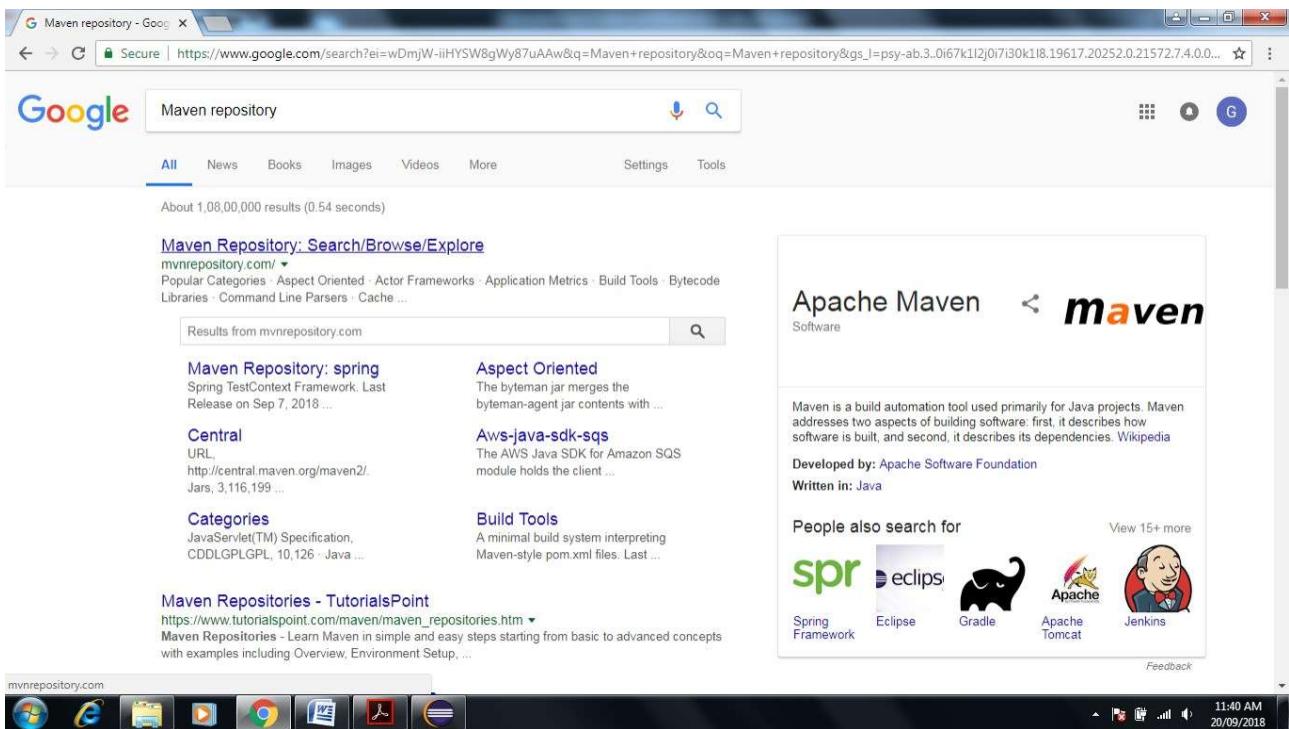
42. it shown some description like <modelVersion>4.0.0</modelVersion>
<groupId> MavenTestReportDemo </groupId>
<artifactId> MavenTestProject </artifactId> <version>0.0.1-SNAPSHOT </version>



43. We add dependencies to pom.xml of Junit and Selenium



44. To add dependency Go to Google.com->Enter Maven repository-> in Search box on Site Enter Junit



45. after Enter keyword Junit inside Seach box then Enter->it shown another Site Maven Repository for Junit Select that site.

A screenshot of a Google search results page. The search term 'Maven repository' is entered in the search bar. The results page shows several links related to Maven, including the official Maven repository and various Maven-related tools like Spring, Eclipse, and Jenkins. On the right side of the results page, there is a large preview window for the 'Apache Maven' website. The Apache Maven website features the Apache logo and a brief description of Maven as a build automation tool for Java projects. It also lists 'People also search for' other tools like Spring, Eclipse, and Jenkins, each accompanied by its respective logo.

A screenshot of a Google search results page for 'Junit site:mvnrepository.com'. The search term 'Junit site:mvnrepository.com' is entered in the search bar. The results page shows several links related to JUnit, including the official Maven repository page for JUnit. This page provides details about JUnit, such as its version (4.12), repositories, license (EPL 1.0), and categories (Testing Frameworks). The page also includes links to the JUnit GitHub repository and a link to the Maven repository page for JUnit.

**46. Click on Maven Repository-JUnit it open another site-
(<https://mvnrepository.com/artifact/junit>)**

The screenshot shows the Maven Repository website at <https://mvnrepository.com/artifact/junit>. The main content area displays the 'Group: JUnit' page. It features a search bar and navigation links for 'Categories', 'Popular', and 'Contact Us'. A sidebar on the left lists 'Indexed Artifacts (12.3M)' and 'Popular Categories' including Aspect Oriented, Actor Frameworks, Application Metrics, Build Tools, Bytecode Libraries, Command Line Parsers, Cache Implementations, Cloud Computing, Code Analyzers, Collections, Configuration Libraries, Core Utilities, Date and Time Utilities, Dependency Injection, and Embedded SQL Databases. The central content area shows two versions of JUnit: '1. JUnit' (78,839 usages) and '2. JUnit' (1,437 usages). Below these are sections for 'Related Books' featuring 'How to use JUnit (2016)' by Van Nguyen and 'Junit with examples (2016)' by Mr Sagar Salunke. The bottom right contains advertisements for LAPP and KnownHost.

47. Click on JUnit-> Open and click on latest version as shown below (here 4.12x)

The screenshot shows the Maven Repository website at <https://mvnrepository.com/artifact/junit/junit>. The main content area displays the 'JUnit' artifact details page. It includes a search bar and navigation links for 'Categories', 'Popular', and 'Contact Us'. The central content area shows the 'License' (EPL 1.0), 'Categories' (Testing Frameworks), 'Tags' (testing, junit), and 'Used By' (78,839 artifacts). A note indicates that the artifact was moved to org.junit.jupiter > junit-jupiter-api. Below this is a table showing usage statistics across various repositories. The table highlights the '4.12' version, which has 34,788 usages in the Central repository on December 2014.

Version	Repository	Usages	Date
4.12	Central	34,788	Dec, 2014
4.12-beta-3	Central	30	Nov, 2014
4.12-beta-2	Central	31	Sep, 2014
4.12-beta-1	Central	31	Jul, 2014
4.11	Central	22,754	Nov, 2012
	Central	22	Oct, 2012

48. Copy above dependency to paste in pom.xml in Maven in Eclipse

The screenshot shows the Maven Repository website at <https://mvnrepository.com/artifact/junit/junit/4.12>. The page displays information about the JUnit 4.12 artifact, including its license (EPL 1.0), categories (Testing Frameworks), organization (JUnit), and various download links. A sidebar advertisement for BOSE SoundLink Revolve speakers is visible. The bottom status bar shows the system tray and the date/time.

```
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
</dependency>
```

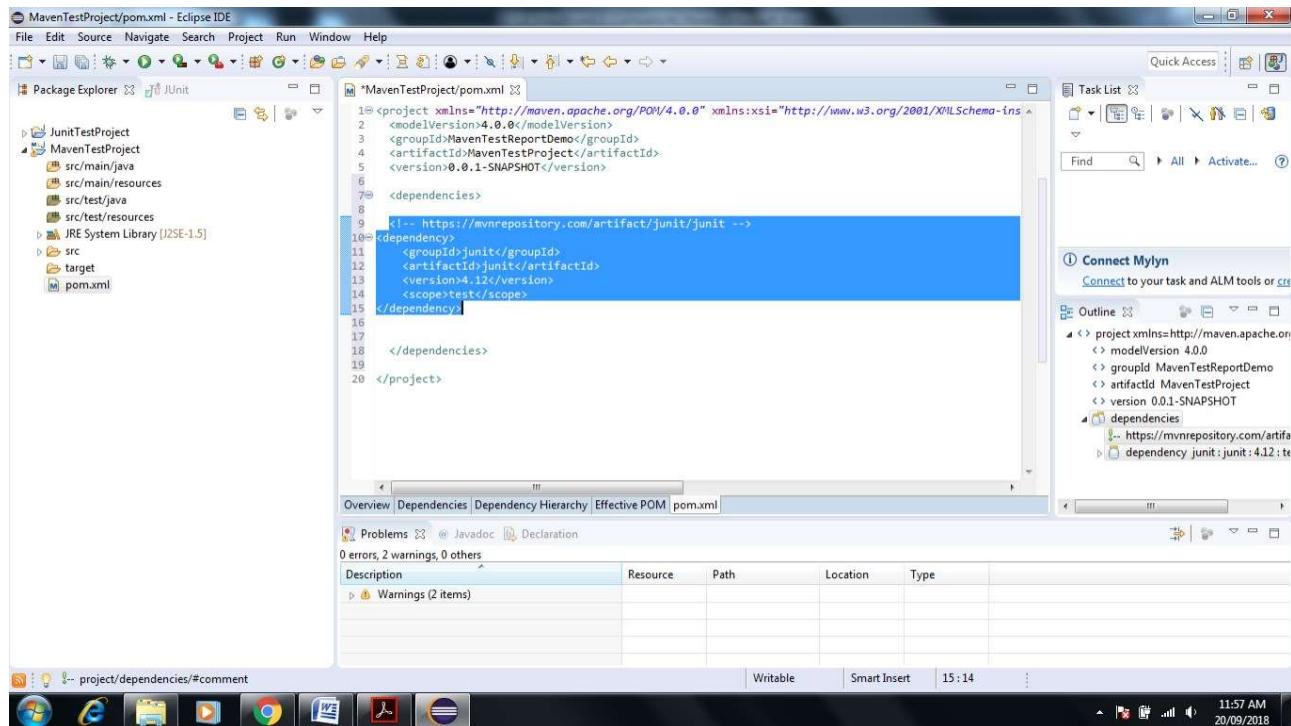
49. Add <dependencies> tag before pasting as shown below

The screenshot shows the Eclipse IDE interface with the MavenTestProject/pom.xml file open in the editor. The code editor shows the XML structure of the POM file, with the cursor positioned at the end of the file (before the final closing tag). The Eclipse interface includes the Package Explorer, Task List, Outline, and Problems view.

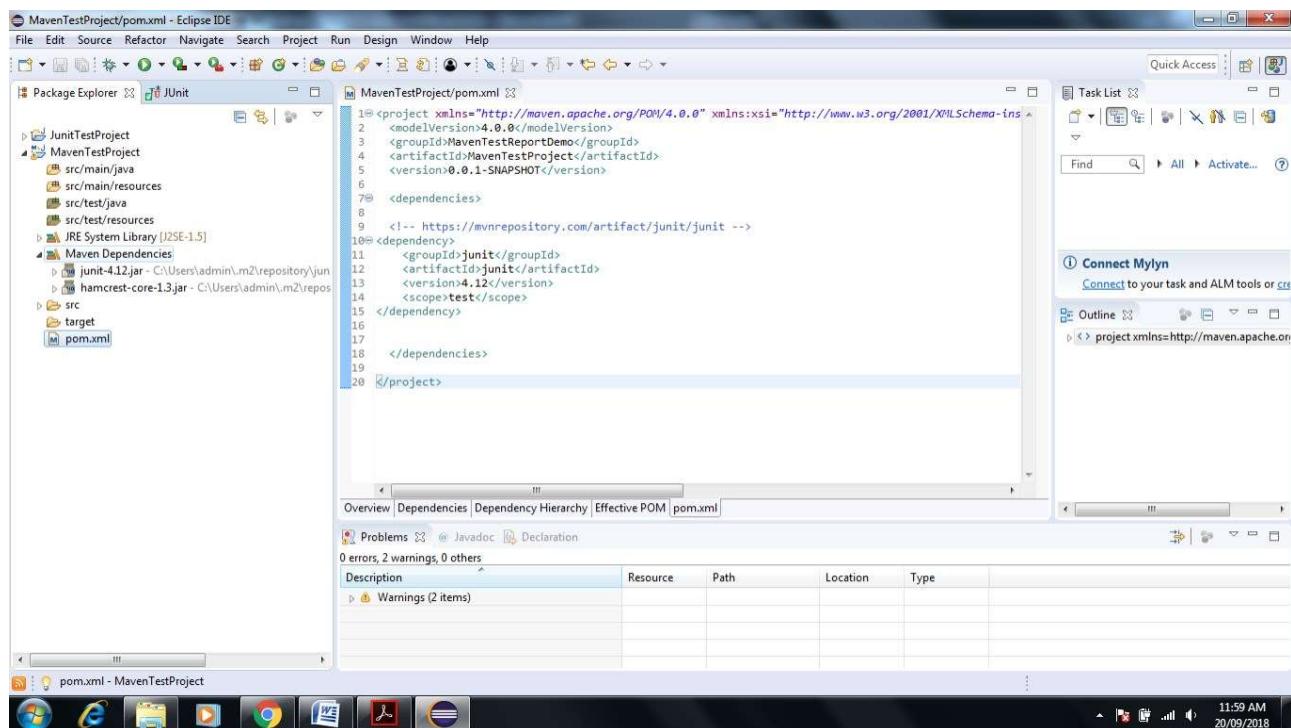
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId> MavenTestReportDemo </groupId>
  <artifactId> MavenTestProject </artifactId>
  <version> 0.0.1-SNAPSHOT </version>
  <dependencies>
  </dependencies>
</project>
```

50. Now Paste the above code in between <dependencies> tag then save pom.xml file <!-- https://mvnrepository.com/artifact/junit/junit -->

```
<dependency> <groupId>junit</groupId> <artifactId>junit</artifactId>
  <version>4.12</version> <scope>test</scope>
</dependency>
```

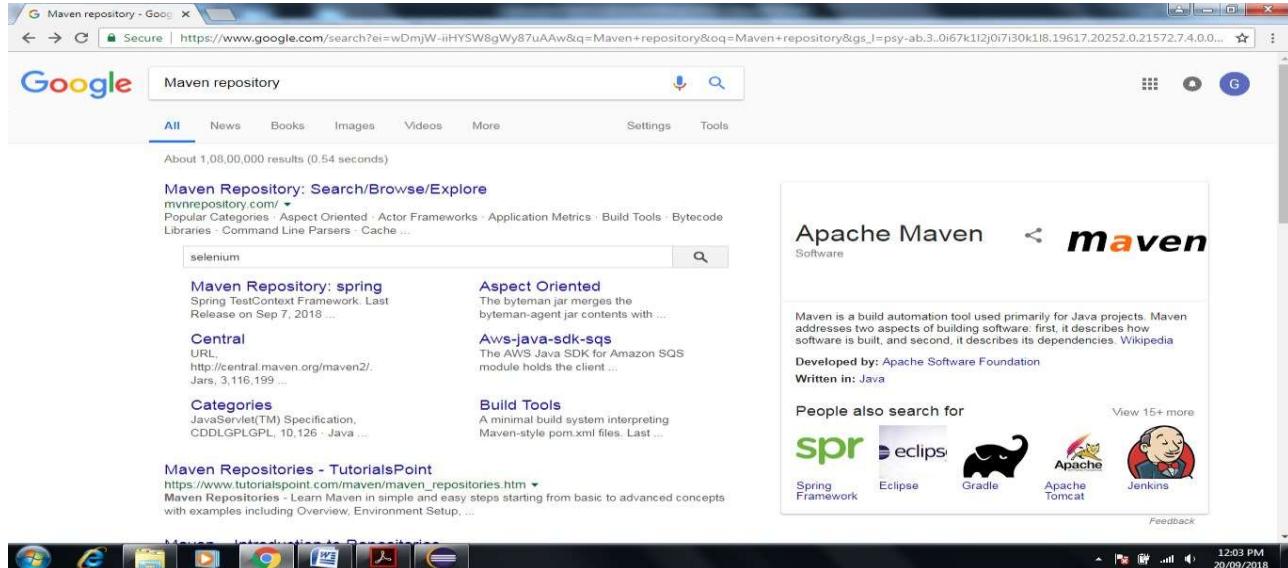


51. Now it gets reflected in Maven by adding Junit jars

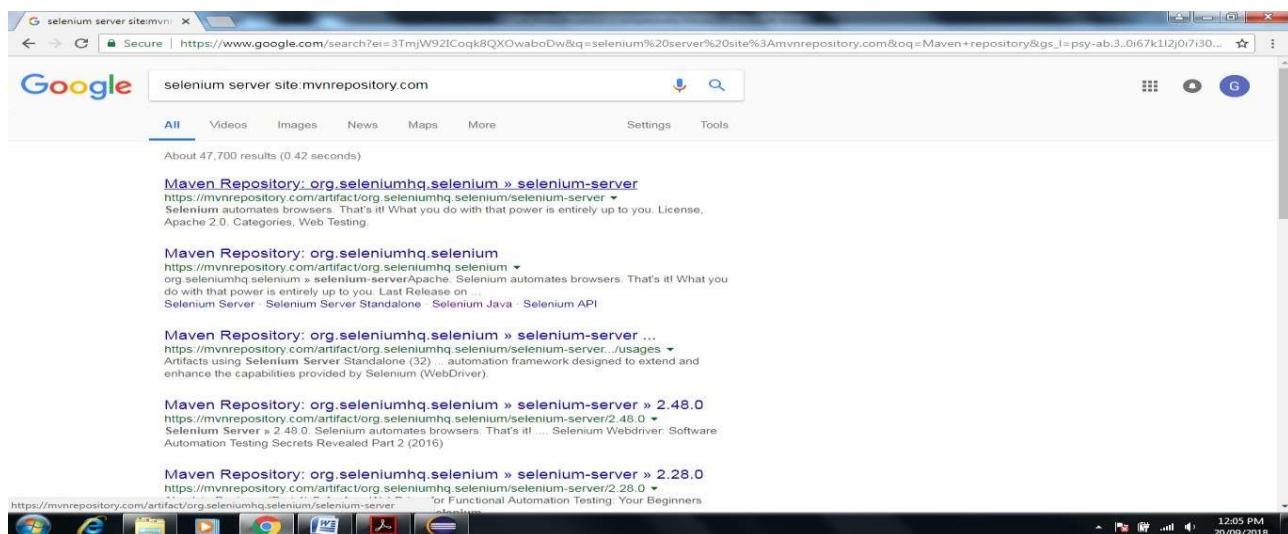


52. Same process can be repeated for Selenium server

Go to Google-> Enter Maven Repository->Enter Selenium Server in Search box->Enter



53. Click on First Link of Website-> Click on latest version



Maven Repository: org.se... Secure | https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-server

MVNREPOSITORY

Indexed Artifacts (12.3M)

Selenium Server

Selenium automates browsers. That's it! What you do with that power is entirely up to you.

License: Apache 2.0

Categories: Web Testing

Tags: testing, selenium, server, web

Used By: 249 artifacts

Central (106) Atlassian 3rdParty (1) Alfresco (1)

Version	Repository	Usages	Date
3.14.x 3.14.0	Central	6	Aug, 2018
3.13.x 3.13.0	Central	8	Jun, 2018
3.12.x 3.12.0	Central	9	May, 2018
3.11.x 3.11.0	Central	13	Mar, 2018
3.10.x 3.10.0	Central	2	Mar, 2018
3.9.x 3.9.1	Central	7	Feb, 2018
3.8.x 3.9.0	Central	1	Feb, 2018
3.8.x 3.8.1	Central	14	Dec, 2017
3.8.x 3.8.0	Central	1	Nov, 2017

ÖLFLEX® UNIPLUS
Single core wires, 1100 voltage grade as per IS 694 for industrial applications.
SHOP NOW

LAPP 30% OFF

12:07 PM 20/09/2018

54. Copy Code in Maven Tab

Maven Repository: org.se... Secure | https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-server/3.14.0

MVNREPOSITORY

Indexed Artifacts (12.3M)

Selenium Server » 3.14.0

Selenium automates browsers. That's it! What you do with that power is entirely up to you.

License: Apache 2.0

Categories: Web Testing

HomePage: http://www.seleniumhq.org/

Date: (Aug 02, 2018)

Files: pom (3 KB) jar (589 KB) View All

Repositories: Central

Used By: 249 artifacts

Maven Gradle SBT Ivy Graal Leiningen Buildr

```
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-server -->
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-server</artifactId>
    <version>3.14.0</version>
</dependency>
```

Include comment with link to declaration

Copied to clipboard!

ÖLFLEX® UNIPLUS SHOP NOW LAPP

KONICA MINOLTA
Remote access for a Secure Print Management in your organisation.

PRINT SOLUTIONS THAT MATTER
PRINT. MANAGE. INNOVATE.

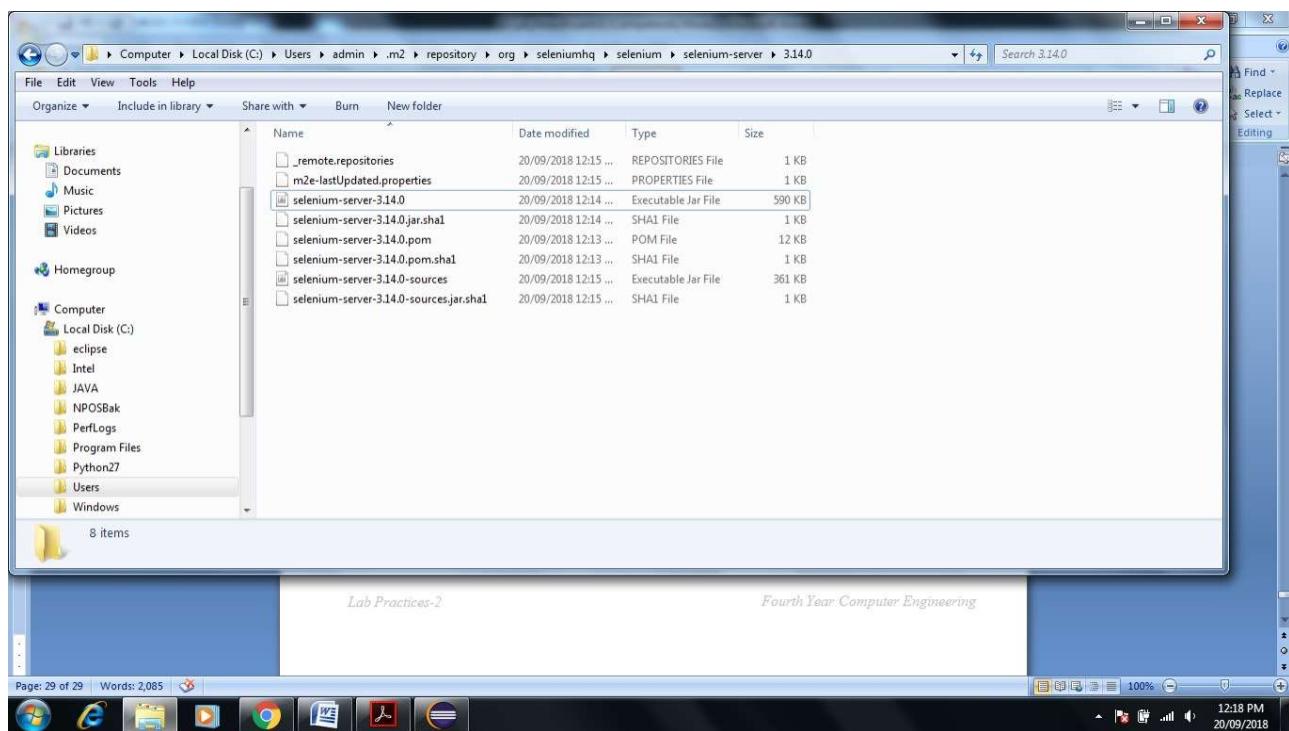
YOSHIDA SAFETY AS Multifunctional Color Printer

12:08 PM 20/09/2018

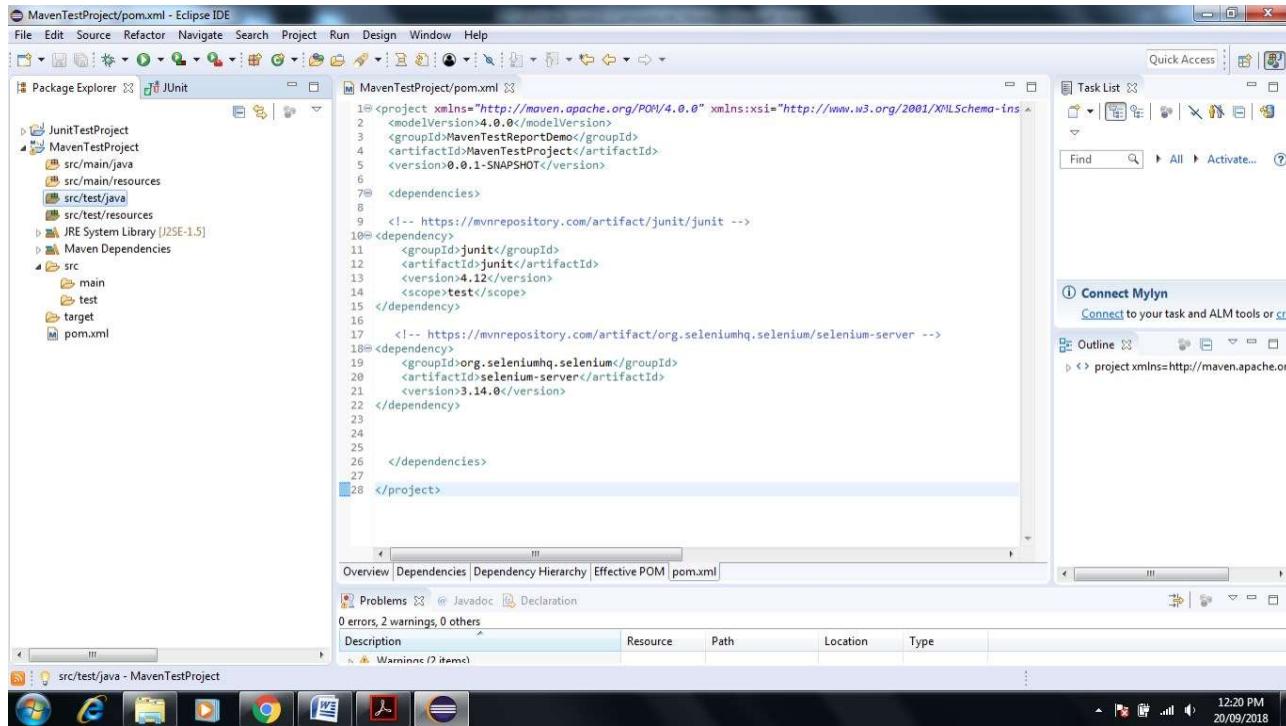
55. Paste in pom.xml file in between <dependencies> tag

```
<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>MavenTestReportDemo</groupId>
  <artifactId>MavenTestProject</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>
    <!-- https://mvnrepository.com/artifact/junit/junit -->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.12</version>
      <scope>test</scope>
    </dependency>
    <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-server -->
    <dependency>
      <groupId>org.seleniumhq.selenium</groupId>
      <artifactId>selenium-server</artifactId>
      <version>3.14.0</version>
    </dependency>
  </dependencies>
</project>
```

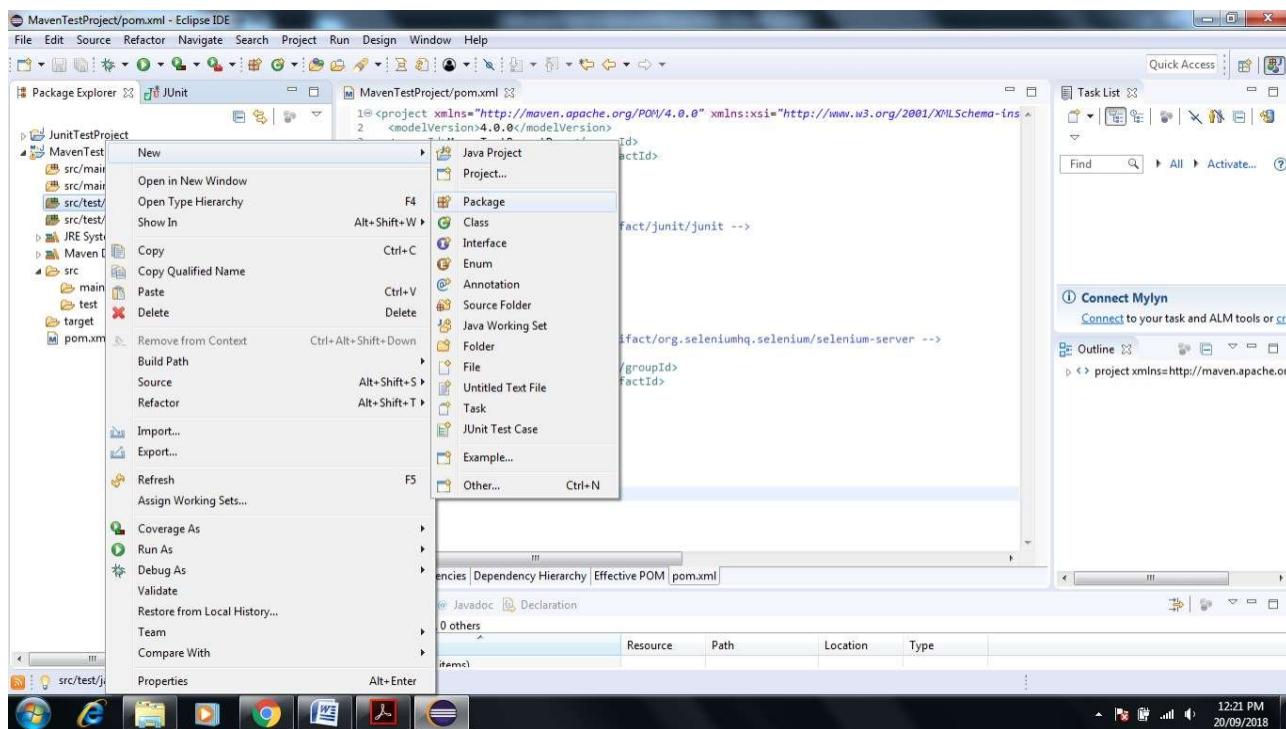
56. Now Go to C:\Users\admin\.m2\repository\org\seleniumhq\selenium\selenium-server\3.14.0 Check the latest selenium server version.



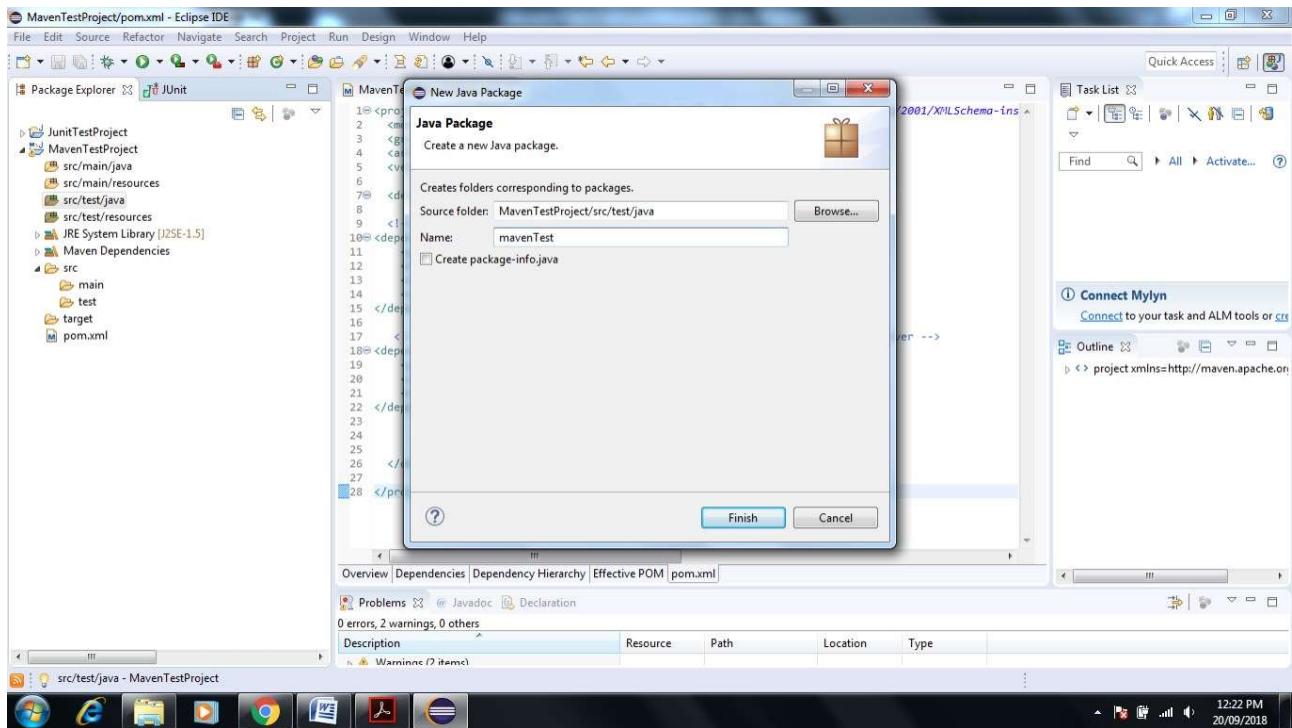
57. Now go to Eclipse -> Click on Maven Test Project->Right Click on src/test/java



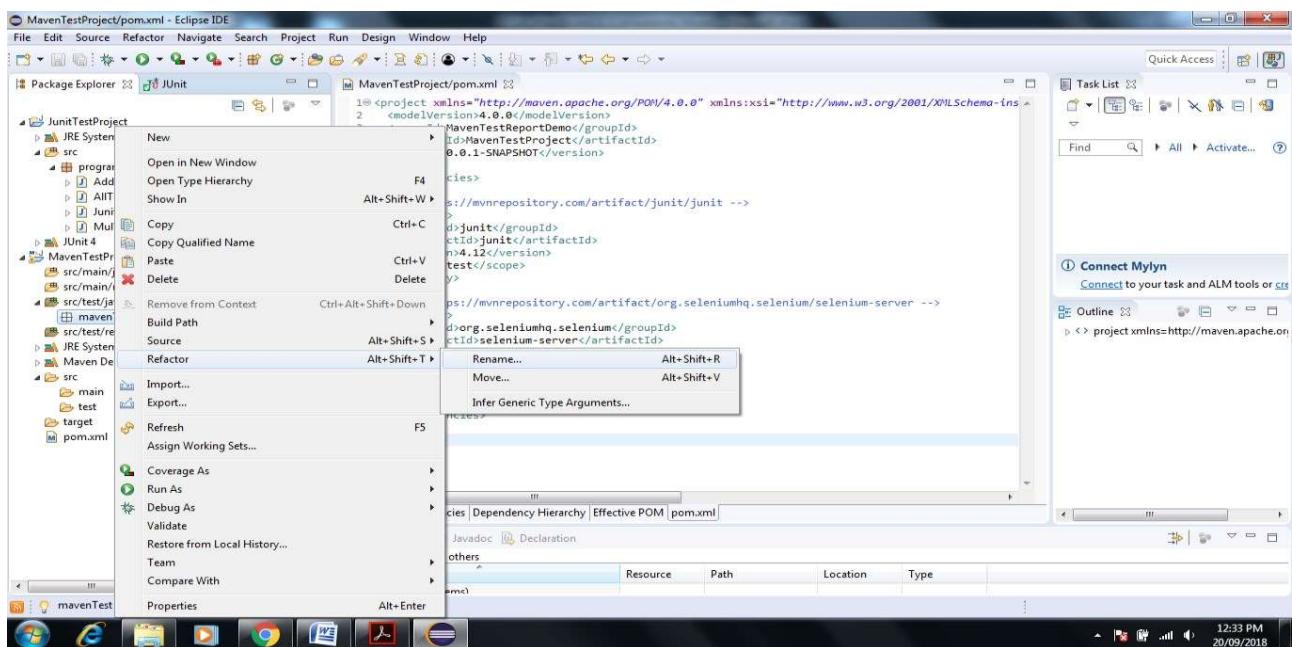
58. Click New->Package



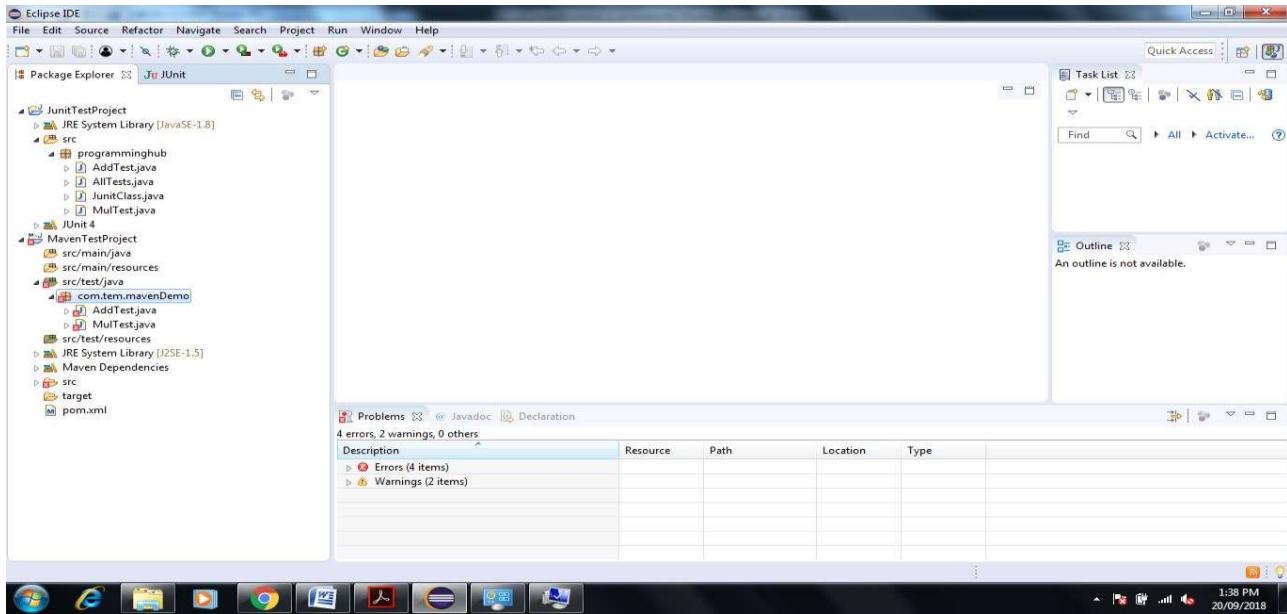
59. Give name to package mavenTest



60. It shows the manvenTest Package under src/test/java folder now rename same by right click on mavenTest Click on Refactor->Rename->give another name com.tem.mavenDemo->Click on ok



61. rename as com.tem.mavenDemo

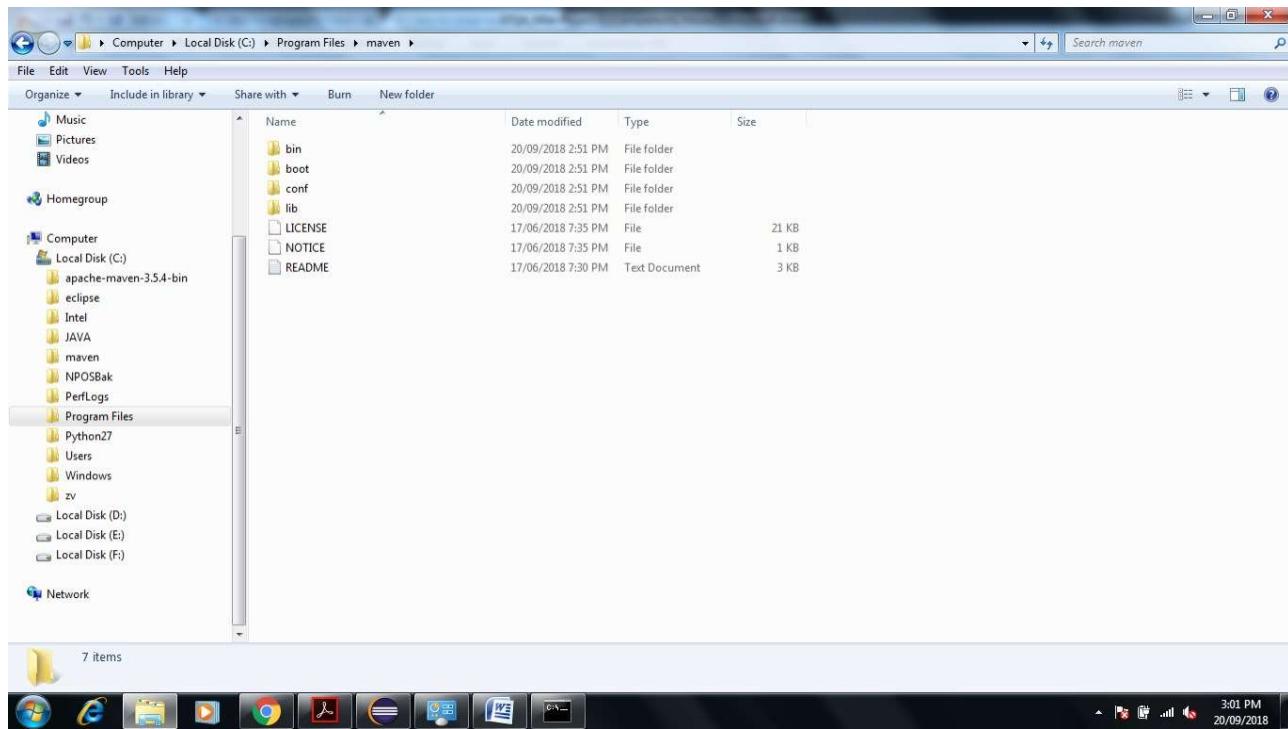


62. Download Apache Maven Select that binary apache-maven-3.5.4-bin

The screenshot shows a web browser window displaying the Apache Maven download page. The URL is maven.apache.org/download.cgi. The page lists several distribution formats for Apache Maven 3.5.4, including tar.gz and zip archives, along with their checksums and signatures. The status bar at the bottom right indicates the time as 1:32 PM and the date as 20/09/2018.

Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.5.4-bin.tar.gz sha512	apache-maven-3.5.4-bin.tar.gz.asc
Binary zip archive	apache-maven-3.5.4-bin.zip sha512	apache-maven-3.5.4-bin.zip.asc
Source tar.gz archive	apache-maven-3.5.4-src.tar.gz sha512	apache-maven-3.5.4-src.tar.gz.asc
Source zip archive	apache-maven-3.5.4-src.zip sha512	apache-maven-3.5.4-src.zip.asc

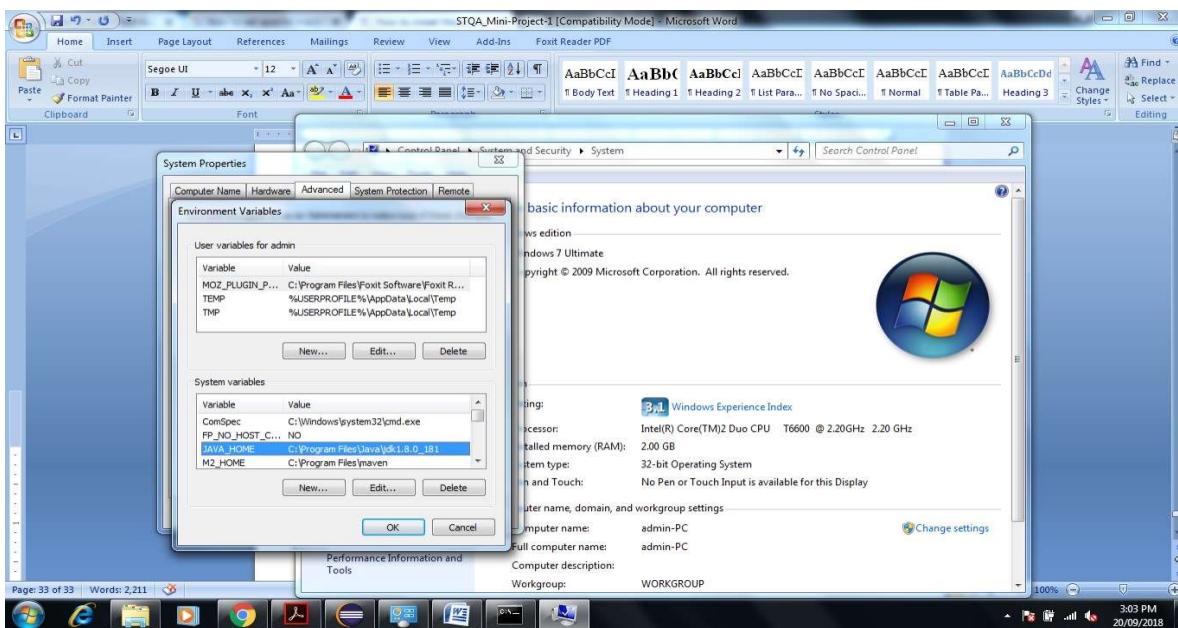
63. after Download->go to Program File->create one folder give name maven-> now extract the downloaded file in maven folder



64. Environment Setup Very Important Steps to Generate Report

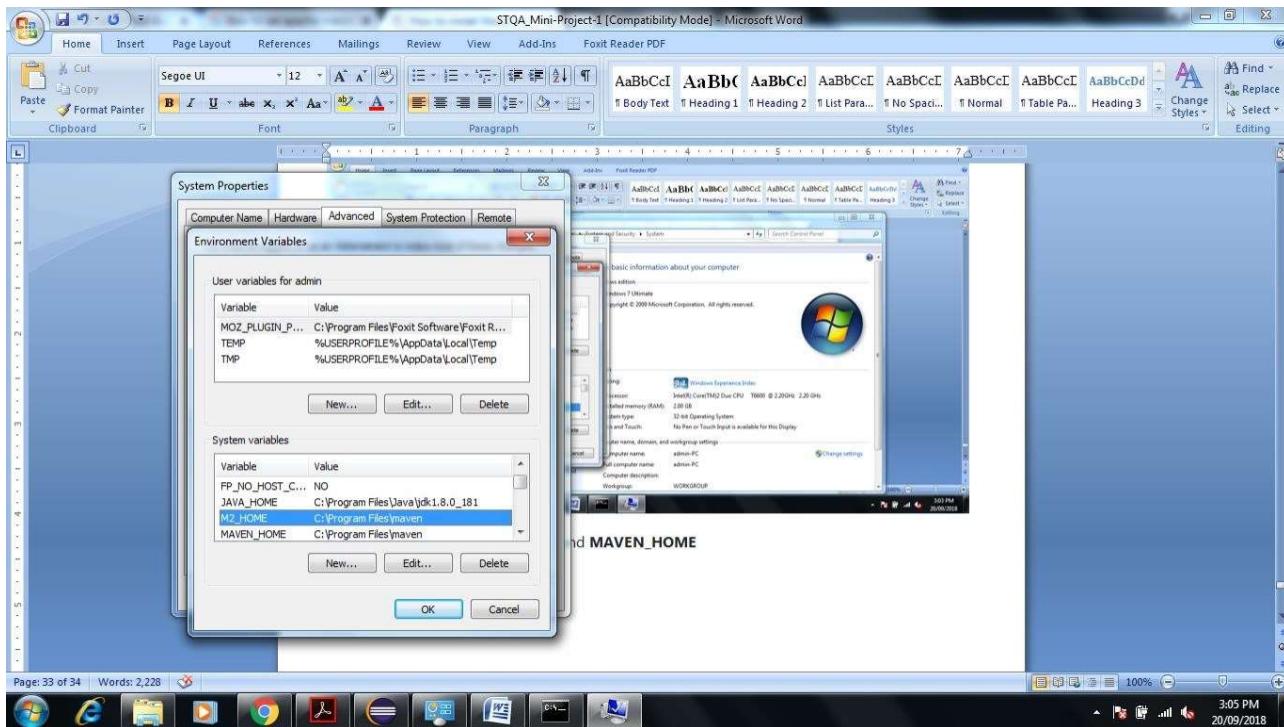
1. JDK and JAVA_HOME

Make sure JDK is installed, and "**JAVA_HOME**" variable is added as Windows environment variable. Our JDK installed in Program File ->JAVA->JDK 1.8.0



2. Set Path of Add M2_HOME and MAVEN_HOME

Create new system variable name M2_HOME and MAVEN_HOME
separately set variable value **C:\Program Files\maven**

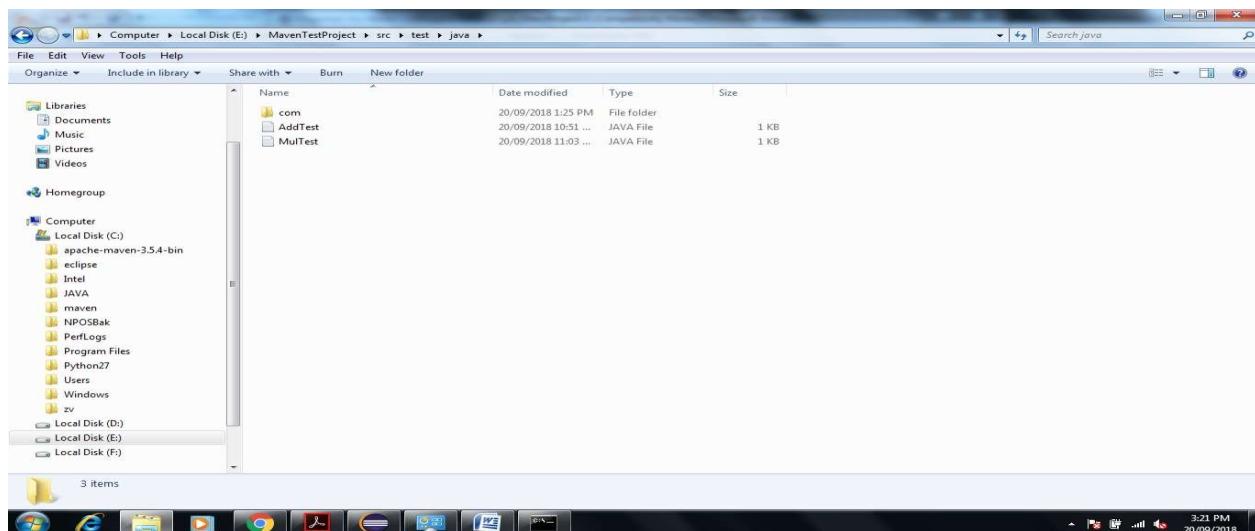


3. Update PATH Variable as per following

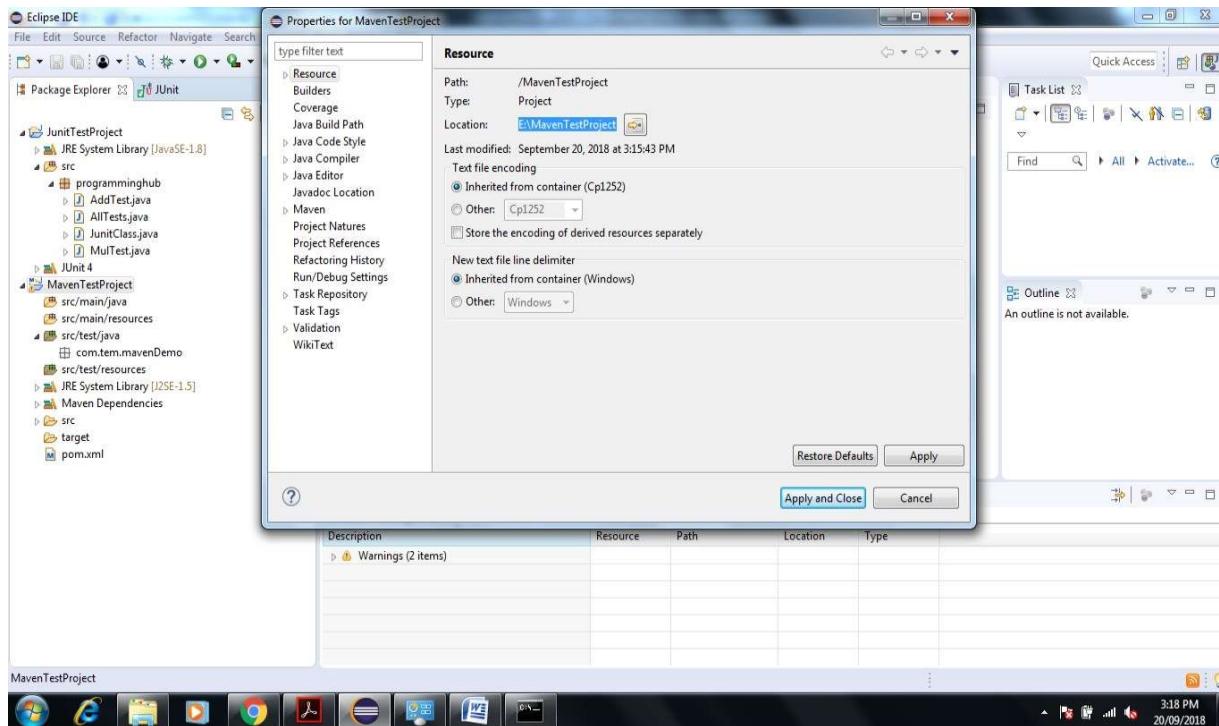
C:\Program Files\maven;%MAVEN_HOME%\bin;%M2_HOME%\bin;

4. Verification

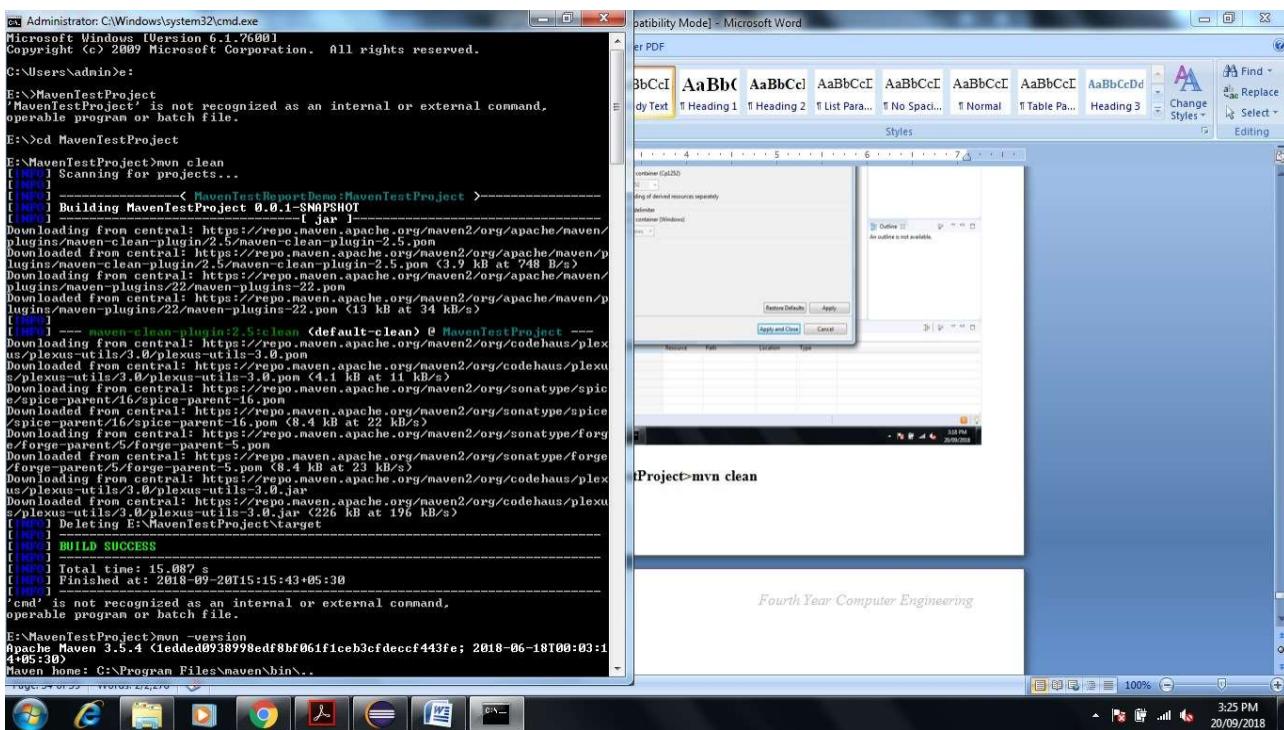
Now copy Previous Created JUnit Test case java file Add Test and Mul Test Paste Externally in **E:\ MavenTestProject\src\test\java**



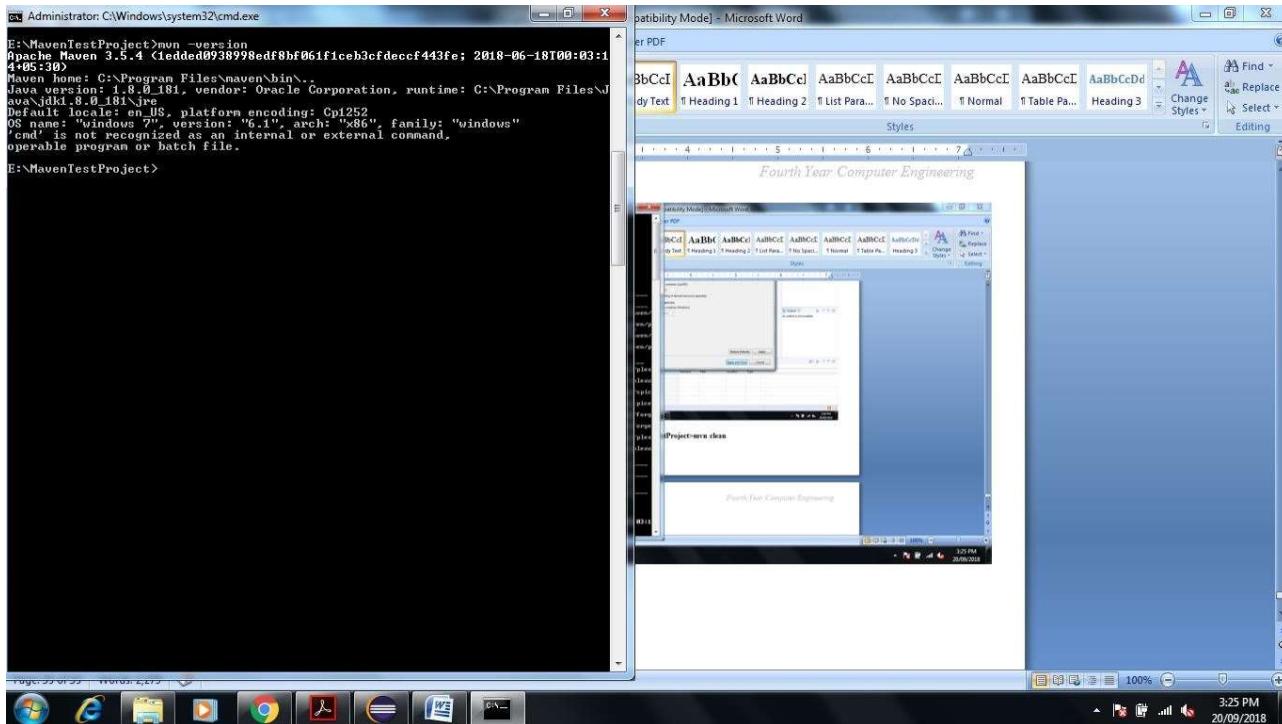
Now Open Eclipse Right Click on MavenTestProject->Properties  Resources  Copy Path of Project Folder



Now go to command prompt  E:\MavenTestProject>mvn clean



Enter E:\MavenTestProject>mvn -version

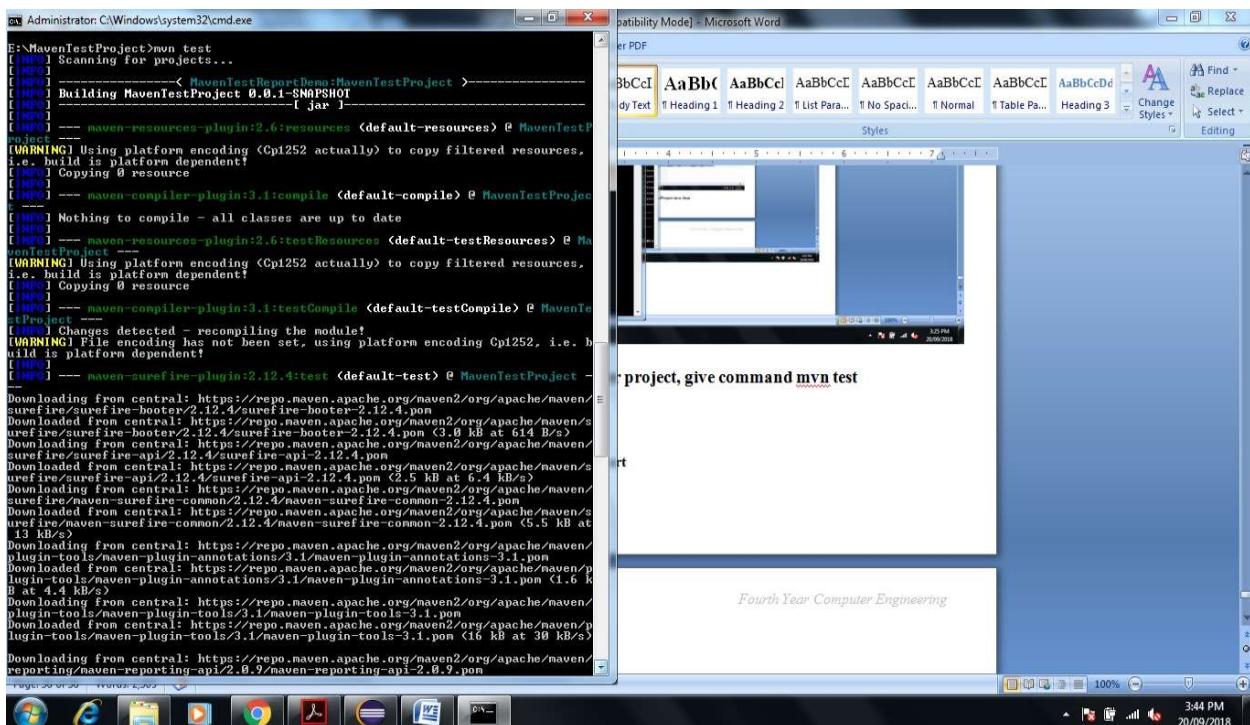


To run test suite or all test cases under project, give command mvn test

Enter E:\MavenTestProject>mvn test

This Command is Used to See the Test Report

You can even run individual test cases. Give command mvn test -Dtesttestcasename



The screenshot shows a Windows desktop environment. In the foreground, there is a Microsoft Word document titled "Compatibility Mode - Microsoft Word" containing the text "THE END*****". In the background, a command prompt window titled "Administrator: C:\Windows\system32\cmd.exe" is open, displaying the output of a Maven test run. The output includes several warning messages about platform encoding and build dependencies, followed by a "BUILD SUCCESS" message and completion details. The taskbar at the bottom shows various icons, and the system tray indicates the date and time as 20/09/2018 3:56 PM.

```
Administrator: C:\Windows\system32\cmd.exe
E:\MavenTestProject>mvn test -DaddTest
[INFO] Scanning for projects...
[INFO] [INFO] < MavenTestReportDemo:MavenTestProject >
[INFO] Building MavenTestProject 0.0.1-SNAPSHOT
[INFO] [INFO] --- [jar]
[INFO] --- maven-resources-plugin:2.6:resources <default-resources> @ MavenTestProject
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,
i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO] --- maven-compiler-plugin:3.1:compile <default-compile> @ MavenTestProject
[INFO] Nothing to compile - all classes are up to date
[INFO] --- maven-resources-plugin:2.6:testResources <default-testResources> @ MavenTestProject
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,
i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO] --- maven-compiler-plugin:3.1:testCompile <default-testCompile> @ MavenTestProject
[INFO] Nothing to compile - all classes are up to date
[INFO] --- maven-surefire-plugin:2.12.4:test <default-test> @ MavenTestProject
[INFO] BUILD SUCCESS
[INFO] Total time: 3.848 s
[INFO] Finished at: 2018-09-20T15:53:21+05:30
[INFO] 
'cmd' is not recognized as an internal or external command,
operable program or batch file.

E:\MavenTestProject>mvn test -Dtest=AllTest
[INFO] Scanning for projects...
[INFO] [INFO] < MavenTestReportDemo:MavenTestProject >
[INFO] Building MavenTestProject 0.0.1-SNAPSHOT
[INFO] [INFO] --- [jar]
[INFO] --- maven-resources-plugin:2.6:resources <default-resources> @ MavenTestProject
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,
i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO] --- maven-compiler-plugin:3.1:compile <default-compile> @ MavenTestProject
[INFO] Nothing to compile - all classes are up to date
[INFO] --- maven-resources-plugin:2.6:testResources <default-testResources> @ MavenTestProject
```

Conclusion:

In this way using JUnit and Maven Automation tool we are Perform Unit Testing and Prepare Test Report of same.

Mini Project No.2

2.1 Title

Create a small web-based application by selecting relevant system environment/platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Narrate scripts in order to perform regression tests. Identify the bugs using Selenium WebDriver and IDEand generate test reports encompassing exploratory testing.

2.2 Problem Definition:

Perform Web testing and identify the bugs using Selenium WebDriver and IDEand generate test reports encompassing exploratory testing.

2.3 Prerequisite: Knowledge of Core Java

2.4 Software Requirements:

Eclipse photon R latest Version, JAVA 1.8, selenium-server-standalone-3.13.0
Chromedriver.exe

2.5 Hardware Requirement:

PIV, 2GB RAM, 500 GB HDD, Lenovo A13-4089Model.

2.6 Learning Objectives:

We are going to learn how Identify the bugs using Selenium WebDriver and IDEand generate test reports encompassing exploratory testing.

2.7 Outcomes:

You are able to Web Testing using Automation Tool like Selenium Web driver and IDE

2.8 TheoryConcepts:

2.8.1 What is Selenium?

Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms.

Selenium is a suite of software tools to automate Web Browsers.

- It is an Open source suite of tools mainly used for Functional and Regression Test Automation.

Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms.

It is quite similar to HP Quick Test Pro (QTP now UFT) only that Selenium focuses on automating web-based applications. Testing done using Selenium tool is usually referred as Selenium Testing.

- **Selenium supports various Operating environments.**

- ✓ MS Windows
- ✓ Linux
- ✓ Macintosh etc...

- **Selenium supports various Browsers.**

- ✓ Mozilla Firefox
- ✓ IE
- ✓ Google Chrome
- ✓ Safari
- ✓ Opera etc...

Note: Selenium IDE supports Mozilla Firefox only.

- **Selenium supports various programming environments to write programs (Test scripts)**

- ✓ Java
- ✓ C#

✓ Python

✓ Perl

✓ Ruby

✓ PHP

2.8.2 History of the Selenium Project

Selenium first came to life in 2004.

- In 2006, Selenium WebDriver was launched at Google.
- In 2008, the whole Selenium team decided to merge Selenium WebDriver with Selenium RC in order to form more powerful tool called Selenium 2.0

✓ **Selenium 1**

(Selenium IDE + Selenium RC + Selenium Grid)

✓ **Selenium 2**

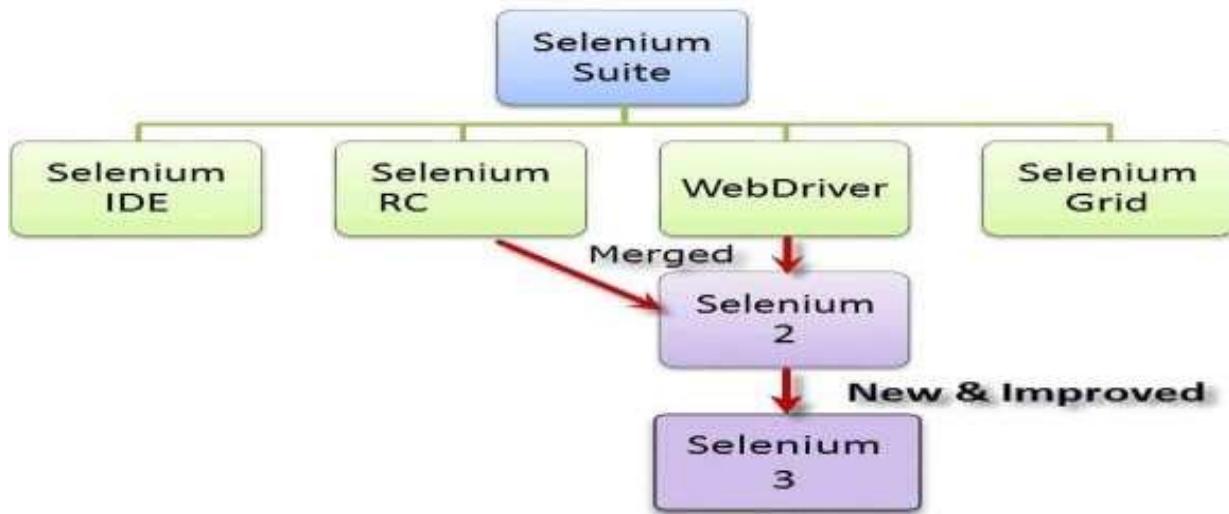
(Selenium IDE + Selenium RC + Selenium WebDriver + Selenium Grid)

2.8.3 Selenium's Tools Suite

Selenium is not just a single tool but a suite of software's, each catering to different testing needs of an organization.

It has four components.

- **Selenium Integrated Development Environment (IDE)**
- **Selenium Remote Control (RC)**
- **WebDriver**
- **Selenium Grid**



2.8.4 Brief Introduction Selenium IDE

It is a Firefox browser plug in, used to create and execute Test cases.

1. Selenium IDE Features:

- Create Test Cases, Test suites (We can Record test cases or type Test steps using element locators and Selenese commands)
- Edit Test Cases
- Execute Test cases, Test suites
- Debug Test Cases.
- Enhance Test Cases
- Export Test cases to other formats (java, ruby etc...)

Note: **selenium IDE Test case default format is .html**

2. Drawbacks of Selenium IDE

- It supports Mozilla Firefox browser only.
- It doesn't support Programming logic/features to enhance Test cases.
- It doesn't support Data Driven Testing.
- It is not suitable for complex test case design.
- No centralized maintenance of Objects/Elements

3. Selenium RC (* Out dated) -Currently, Selenium RC is still being developed but only in maintenance mode.

4. Selenium WebDriver

- ✓ It is a Programming interface to create and execute Test cases.

Selenium IDE has IDE but doesn't have Programming interface

- ✓ Selenium WebDriver has Programming interface but doesn't have IDE
- ✓ It communicates Directly to the browser.
- ✓ No need of Separate Server such as RC Server
- ✓ UFT/QTP has both IDE as well as Programming interface
- ✓ Faster Execution than IDE & RC

• Selenium WebDriver supports various programming environments to write programs.

- ✓ Java,
- ✓ C#
- ✓ Perl

- ✓ Python
 - ✓ Ruby
 - ✓ PHP
- Using Element/Object locators/properties and Webdriver Methods we can create and execute Test cases.
 - Selenium Webdriver supports various browsers to create and execute test case/test script/test

Note: **Browser driver varies from one browser to another.**

- **Selenium WebDriver supports various operating environments**
 - ✓ MS Windows
 - ✓ Linux Macintosh etc...

Drawback of Selenium WebDriver

- It doesn't generate detailed Test Reports.
- No centralized maintenance of Object/elements
- It require Programming Knowledge
- cannot support the readily new browser
- Installation is More Complicated than Selenium IDE
- No built-in mechanism for logging runtime message

5. Selenium Grid

- Selenium Grid is used to execute tests across multiple browsers, operating environments and machines in parallel.

- Selenium Grid 2 supports Selenium RC Tests as well as Selenium WebDriver Tests.
- i) Selenium WebDriver to create Test cases using element locators and Webdriver methods.
- ii) Java Programming to enhance test cases.
- iii) TestNG Framework to group test cases, execute test batches and generate detailed test reports.
- Features:**
- Enables simultaneous running of tests in multiple browsers and environments. • Saves time enormously.
 - Utilizes the **hub-and-nodes** concept. The hub acts as a central source of Selenium commands to each node connected to it.

Note on Browser and Environment Support

- Because of their architectural differences, Selenium IDE, Selenium RC, and WebDriver support different sets of browsers and operating environments.

	Selenium IDE	WebDriver
Browser Support	Mozilla Firefox	Internet Explorer versions 6 to 11, both 32 and 64-bit Microsoft Edge version 12.10240 & above(Partial support some functionalities under development) Firefox 3.0 and above Google Chrome 12.0 and above Opera 11.5 and above Android – 2.3 and above for phones and tablets (devices & emulators) iOS 3+ phones (devices & emulators) and 3.2+ for tablets (devices & emulators) HtmlUnit 2.9 and above
Operating System	Windows, Mac, OS X, Linux	All operating systems where the browsers above can run.

Note: Selenium WebDriver is termed as the successor of Selenium RC which has been deprecated & officially announced by SeleniumHQ.

Tool	Why Choose?
Selenium IDE	<ul style="list-style-type: none"> • To learn about concepts on automated testing and Selenium • Selenium commands such as type, open, clickAndWait assert, verify, etc. • Locators such as id, name, xpath, css selector, etc. • Executing customized JavaScript code using runScript • Exporting test cases in various formats. • To create simple test cases and test suites that you can export later to RC or WebDriver. • To test a web application against Firefox only.
Selenium RC	<ul style="list-style-type: none"> • To design a test using a more expressive language than Selenese • To run your test against different browsers (Except HtmlUnit) on different operating systems. • To deploy your test across multiple environments using Selenium Grid. • To test your application against a new browser that supports JavaScript. • To test web applications with complex AJAX-based scenarios.
WebDriver	<ul style="list-style-type: none"> • To use a certain programming language in designing your test case. • To test applications that are rich in AJAX-based functionalities. • To execute tests on the HtmlUnit browser. • To create customized test results.
Selenium Grid	<ul style="list-style-type: none"> • To run your Selenium RC scripts in multiple browsers and operating systems simultaneously. • To run a huge test suite, that needs to complete in the soonest time possible.

2.8.6 Advantages of Selenium

- i) It is an Open source Software.
- ii) It supports various Operating environments (Windows, Linux, Mac etc...)
- iii) It supports various browsers (IE, Mozilla Firefox, Chrome, safari, Opera etc...)
- iv) It supports various programming environments (Java, Perl, Python, Ruby and PHP)
- v) It supports parallel Test execution.
- vi) It uses less Hardware resources.

2.8.7 Disadvantages of Selenium

- i) It supports Web based Applications only.
- ii) No reliable support from anybody.
- iii) No centralized maintenance of Elements/objects
- iv) Difficult to setup environment.
- v) Difficult to use.
- vi) Limited support for Image based testing.
- vii) New features may not work properly.
- viii) No other tool integration for test management & No built in Reporting facility.

2.8.8 Selenium Versus UFT

Selenium	UFT/QTP
Open Source	Vendor tool, License is required
Supports various OS Environments	MS Windows only
Supports various Programming Environments	VB Script only
No object Repositories	Local and Shared object Repositories
No built-in Reporting feature	Built-in reporting feature
Selenium WebDriver has no IDE and Selenium IDE has no programming Interface	UFT has both IDE and Programming Interface
Uses less Hardware resources	Uses more Hardware resources
Difficult to setup environment and use	Easy to setup and use
Limited support for Image Testing	Rich support for Image Testing
No Reliable support	Support from HP
No other tool integration for Test Management	UFT can be integrated with ALM/QC for Test Management.
New features may not work properly	New features will work properly
No Add ins for supporting Application Environments	Add ins are required for supporting Application environments
Supports Web Applications only	Supports Desktop and Web Applications
No Authorized Certification	Authorized Certification Program

2.8.9 What is TestNG?

TestNG is a powerful testing framework, an enhanced version of JUnit which was in use for a long time before TestNG came into existence. NG stands for 'Next Generation'.

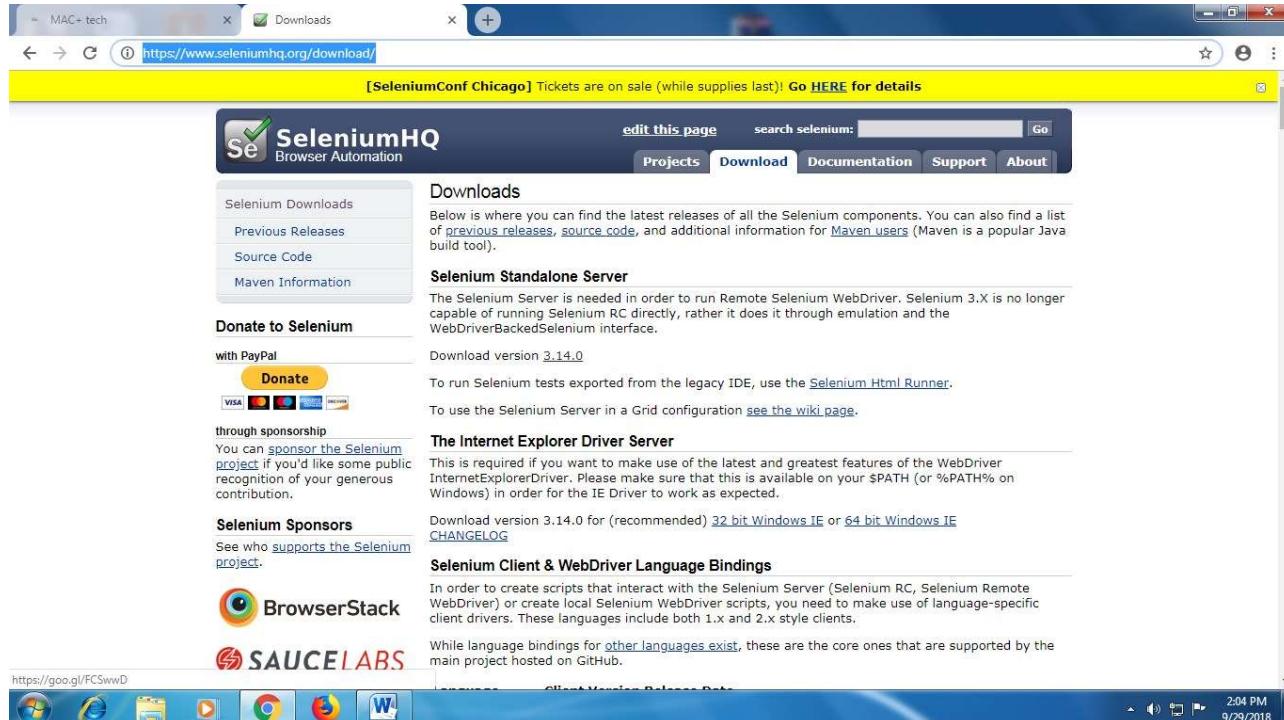
TestNG framework provides the following features –

- Annotations help us organize the tests easily.
- Flexible test configuration.
- Test cases can be grouped more easily.
- Parallelization of tests can be achieved using TestNG.
- Support for data-driven testing.
- Inbuilt reporting.

2.8.9 Step by Step Tutorial

1. First of Download Latest Eclipse java photon-R version.
2. Download latest selenium-server-standalone-3.13.0 jar File from following link

<https://www.seleniumhq.org/download/> here on site 3.14.0 version is latest



3. Download and Extract **Chromedriver.exe** for windows on any drive of computer.

<http://chromedriver.chromium.org/downloads> here 2.42 is latest version

The screenshot shows a web browser window with the URL <https://chromedriver.storage.googleapis.com/2.42/>. The page title is "ChromeDriver - WebDriver for Chrome". On the left, there's a sidebar with links like CHROMEDRIVER, CAPABILITIES & CHROMEOPTIONS, CHROME EXTENSIONS, CHROMEDRIVER CANARY, CONTRIBUTING, DOWNLOADS, GETTING STARTED (Android, ChromEOS), LOGGING (Performance Log), MOBILE EMULATION, and NEED HELP? (Chrome doesn't start or crashes immediately, Chromedriver crashes, Clicking issues, DevTools window keeps closing, Operation not supported when using remote debugging). The main content area is titled "Latest Release: ChromeDriver 2.42" and states "Supports Chrome v68-70". It includes a section titled "Changes include:" with a bulleted list of fixes and improvements. At the bottom of the page, there's a table showing file details for the download:

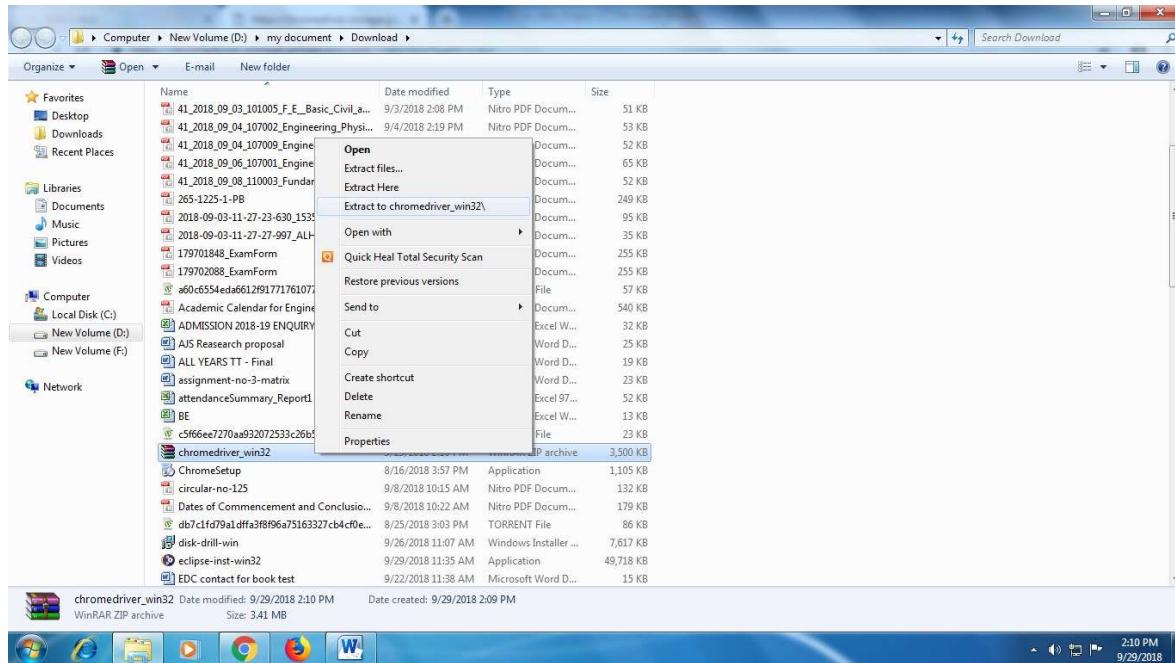
Name	Last modified	Size	ETag
Parent Directory		-	-
chromedriver_linux64.zip	2018-09-13 19:30:37	3.85MB	acfcc29fb03df9e913ef4c360a121ad1
chromedriver_mac64.zip	2018-09-13 18:14:11	5.75MB	3fc0e4a97cbf2c8c2a9b824d95e25351
chromedriver_win32.zip	2018-09-13 21:11:33	3.42MB	28d91b31311146250e7ef1afbcd6d026
notes.txt	2018-09-13 21:23:09	0.02MB	18bdf6fc9f9d8dd668fa444b77d06bdd

The screenshot shows a web browser window with the URL <https://chromedriver.storage.googleapis.com/index.html?path=2.42/>. The page title is "Index of /2.42/". It displays a table of files and their details, identical to the one in the previous screenshot. The table header is:

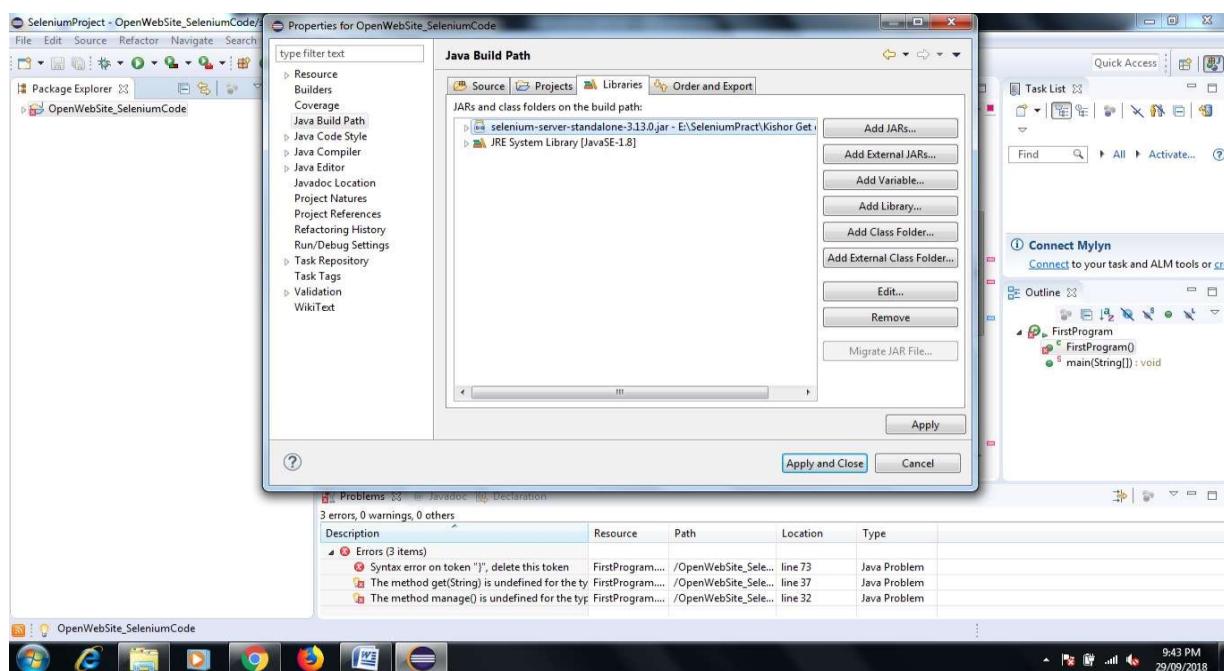
Name	Last modified	Size	ETag
Parent Directory		-	-
chromedriver_linux64.zip	2018-09-13 19:30:37	3.85MB	acfcc29fb03df9e913ef4c360a121ad1
chromedriver_mac64.zip	2018-09-13 18:14:11	5.75MB	3fc0e4a97cbf2c8c2a9b824d95e25351
chromedriver_win32.zip	2018-09-13 21:11:33	3.42MB	28d91b31311146250e7ef1afbcd6d026
notes.txt	2018-09-13 21:23:09	0.02MB	18bdf6fc9f9d8dd668fa444b77d06bdd

The screenshot shows a web browser window with the URL https://chromedriver.storage.googleapis.com/2.42/chromedriver_win32.zip. The page title is "Index of /2.42/". The address bar at the top shows the full URL. The browser interface includes standard navigation buttons (back, forward, search) and a taskbar at the bottom.

4. After Download Extract same on any Drive here I m extract on D Drive and my path of that exe file is D:\my document\Download\chromedriver_win32



5. Now Open Eclipse IDE-----> Create Java Project-> Right Click Project Name→Properties→ Java Build Path→Libraries→Add External JAR→ add selenium-server-standalone-3.13.0 jar →Apply and Close.

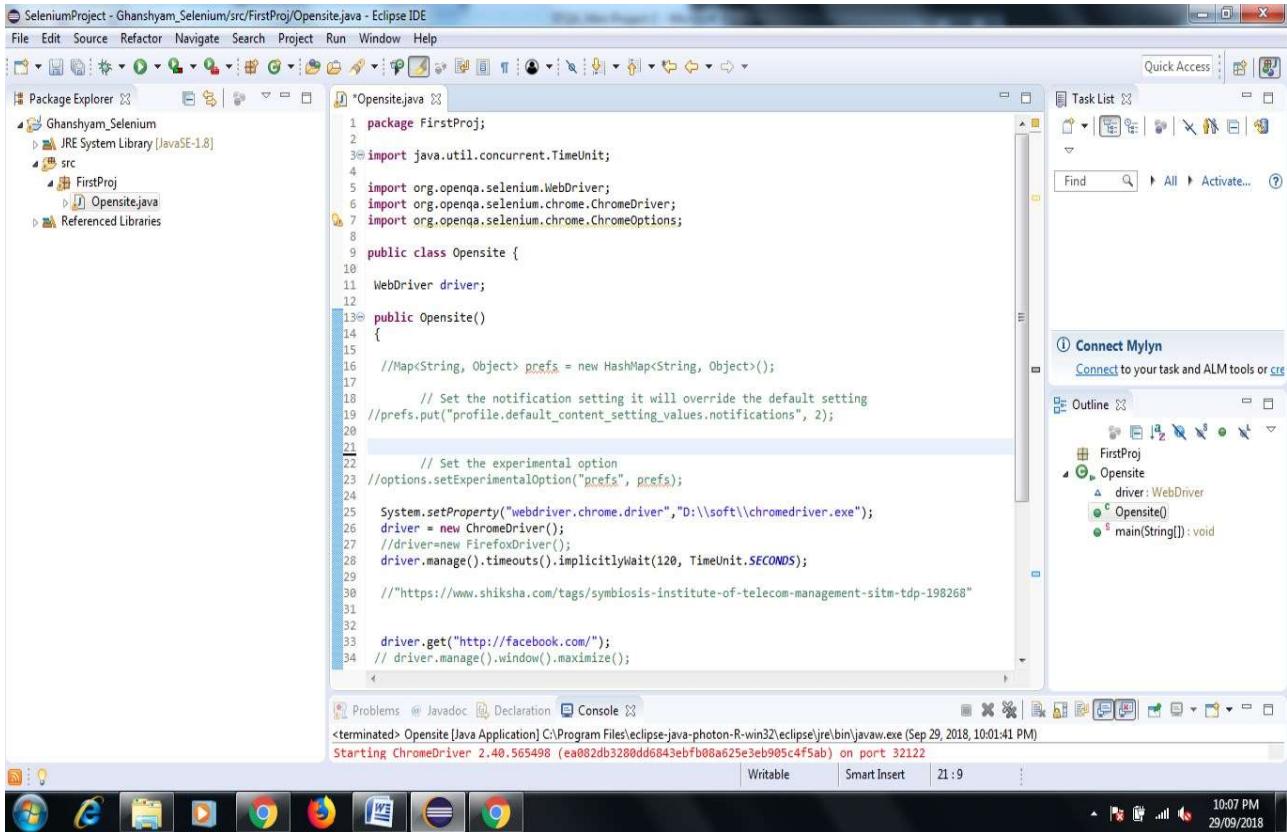


6. Now Want Open the chrome browser with facebook.com page via Selenium

Web driver Java Coding so here we need to write Java Code in Class file which we already created

7. Here in My Program I Create **Ghanshyam_Selenium Java Project**

Folder name and **Opensite.java** is my class file so write java code in this class file



The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** SeleniumProject - Ghanshyam.Selenium/src/FirstProj/Opensite.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbars:** Standard, Java, Java Editor, Java Outline, Java Problems, Java Declaration, Java Console.
- Left Sidebar:** Package Explorer (Ghanshyam.Selenium, src, FirstProj, Opensite.java), JRE System Library [JavaSE-1.8], Referenced Libraries.
- Central Area:** *Opensite.java* (Content Editor) showing the following Java code:

```
1 package FirstProj;
2
3 import java.util.concurrent.TimeUnit;
4
5 import org.openqa.selenium.WebDriver;
6 import org.openqa.selenium.chrome.ChromeDriver;
7 import org.openqa.selenium.chrome.ChromeOptions;
8
9 public class Opensite {
10
11     WebDriver driver;
12
13     public Opensite() {
14
15         //Map<String, Object> prefs = new HashMap<String, Object>();
16
17         // Set the notification setting it will override the default setting
18         //prefs.put("profile.default_content_setting_values.notifications", 2);
19
20
21         // Set the experimental option
22         //options.setExperimentalOption("prefs", prefs);
23
24         System.setProperty("webdriver.chrome.driver", "D:\\soft\\chromedriver.exe");
25         driver = new ChromeDriver();
26         //driver=new FirefoxDriver();
27         driver.manage().timeouts().implicitlyWait(120, TimeUnit.SECONDS);
28
29         //"https://www.shiksha.com/tags/symbiosis-institute-of-telecom-management-sitm-tdp-198268"
30
31
32
33         driver.get("http://facebook.com");
34         // driver.manage().window().maximize();
```
- Right Sidebar:** Task List, Find, Connect Mylyn, Outline (FirstProj, Opensite, driver: WebDriver, Opensite(), main(String[]): void).
- Bottom Status Bar:** Problems, Declaration, Console, Writable, Smart Insert, 21:9, 10:07 PM, 29/09/2018.

8. Right click on java program, select Run As and > "Java Application". After Code Successfully Run now see the output in Console Prompt

SeleniumProject - Ghanshyam_Selenium/src/FirstProj/OpenSite.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

Ghanshyam_Selenium
JRE System Library [JavaSE-1.8]
src
FirstProj
OpenSite.java
Referenced Libraries

OpenSite.java

```
1 package FirstProj;
2
3 import java.util.concurrent.TimeUnit;
4
5 import org.openqa.selenium.WebDriver;
6 import org.openqa.selenium.chrome.ChromeDriver;
7 import org.openqa.selenium.chrome.ChromeOptions;
```

Problems Declaration Console

<terminated> OpenSite [Java Application] C:\Program Files\eclipse-juno-R-win32\eclipse\jre\bin\javaw.exe (Sep 29, 2018, 10:01:41 PM)
Starting ChromeDriver 2.40.565498 (ea082db3280dd6843ebfb08a625e3eb905c4f5ab) on port 32122
Only local connections are allowed.
[1538238724,192][WARNING]: Timed out connecting to Chrome, retrying...
Sep 29, 2018 10:02:06 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS

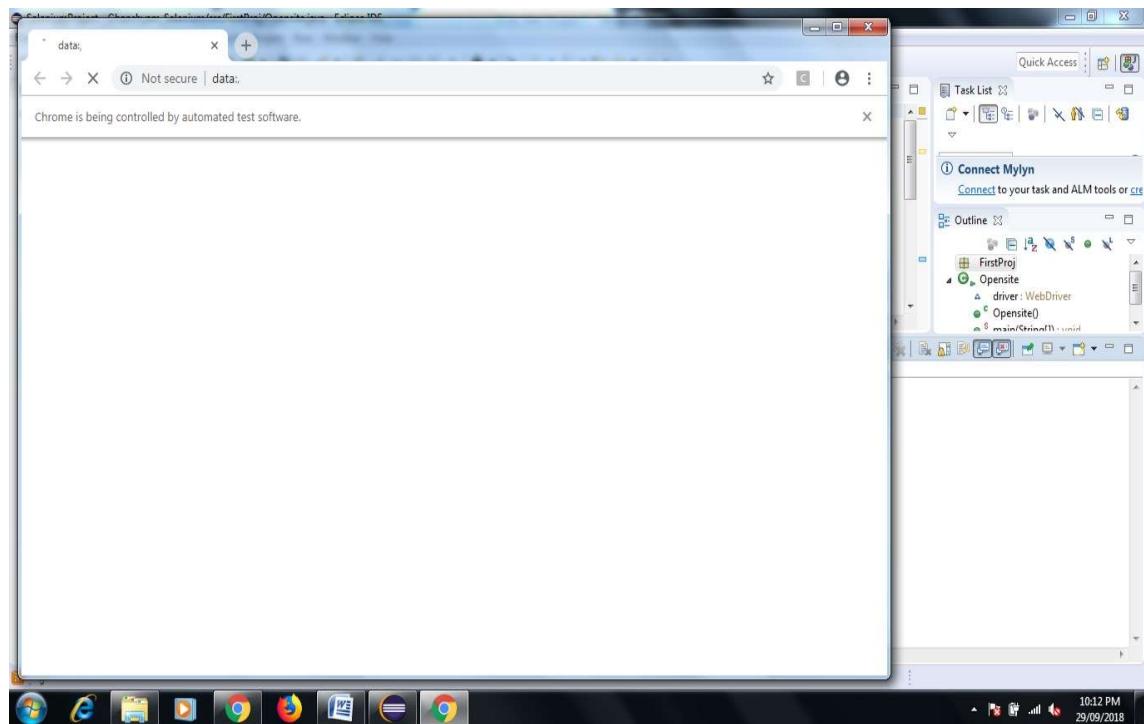
Writable Smart Insert 21 : 9

10:08 PM 29/09/2018

9. Now Your browser Open Automatically it shown data;

10. one more message display chrome is being controlled by Automated Test Software

It mean we open chrome browser and facebook.com page by selenium web driver java code Successfully.



Module-2 In This Project I want to collect all Rating Feedback related our college available on website Justdial.com

1. Create One Java Project Folder Give Name **Review Demo** Now Create One Class File give name **Practo.com**

2. Right Click Project Name **Properties** **Java Build Path** **Libraries** **Add External JAR** **add selenium-server-standalone-3.13.0 jar** **Apply and Close.**

3. Similarly Copy the Chromedriver.exe file on my D Drive D:/soft/chromedriver.exe in this way. 4. Now Start right writing code for Practo.java to Collect all review of any Hospital.

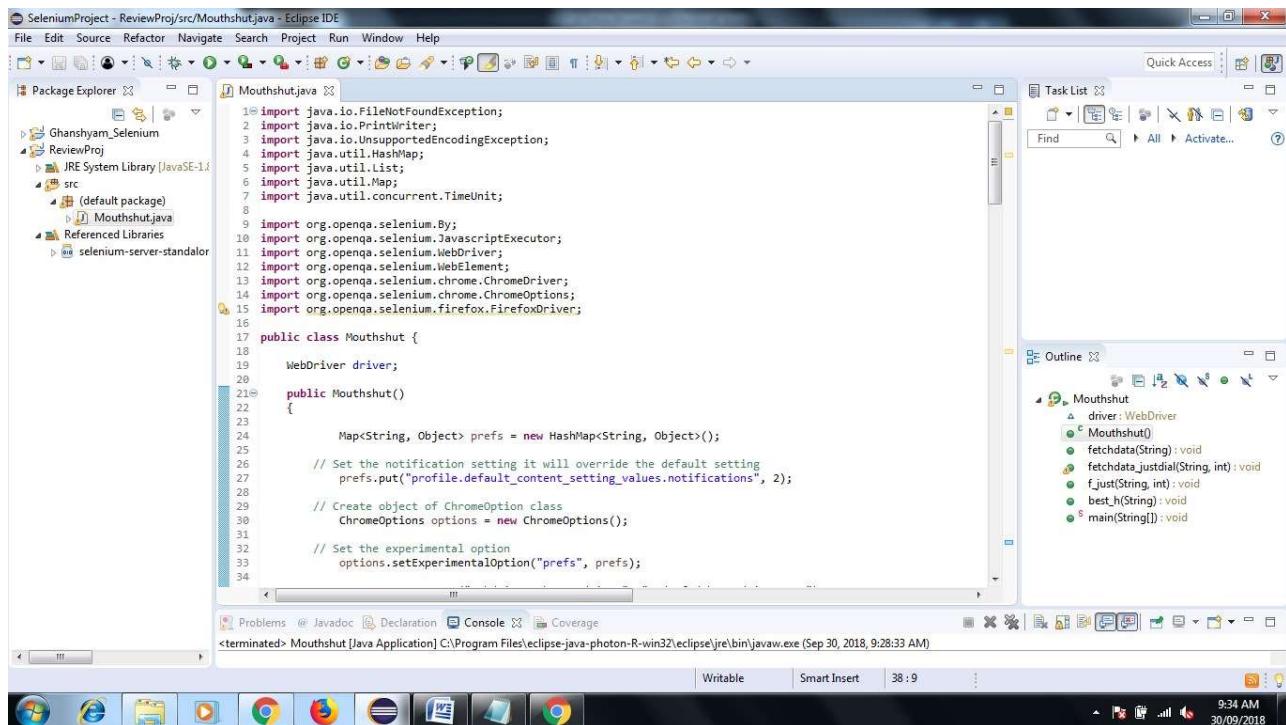
Here I want search all feedback of **SNJB-s-Late-Sau-Kantabai-Bhavarlalji-Jain-College-Of-Engineering-Neminagar-Chandwad**

Link of Feedback of All Customer-

https://www.justdial.com/Nashik/SNJB-s-Late-Sau-Kantabai-Bhavarlalji-Jain-College-Of-Engineering-Neminagar-Chandwad/0253PX253-X253-151019113056-B2P9_BZDET/reviews/page-6

5. Now to save Feedback of all pages here I create one text file give name **C:\Users\admin\eclipse-workspace\SeleniumProject\ReviewProj\Snjb.txt**

Now First of All Execute Code here my file name Mouthshut.java Mouthshut.com is website like Justdialcom



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with packages **Ghanshyam_Selenium**, **ReviewProj**, and **src**. The **Mouthshut.java** file is selected in the **Mouthshut** package.
- Mouthshut.java Content:** The code is as follows:

```
1 import java.io.FileNotFoundException;
2 import java.io.PrintWriter;
3 import java.io.UnsupportedEncodingException;
4 import java.util.HashMap;
5 import java.util.List;
6 import java.util.Map;
7 import java.util.concurrent.TimeUnit;
8
9 import org.openqa.selenium.By;
10 import org.openqa.selenium.JavascriptExecutor;
11 import org.openqa.selenium.WebDriver;
12 import org.openqa.selenium.WebElement;
13 import org.openqa.selenium.chrome.ChromeDriver;
14 import org.openqa.selenium.chrome.ChromeOptions;
15 import org.openqa.selenium.firefox.FirefoxDriver;
16
17 public class Mouthshut {
18
19     WebDriver driver;
20
21     public Mouthshut() {
22
23
24         Map<String, Object> prefs = new HashMap<String, Object>();
25
26         // Set the notification setting it will override the default setting
27         prefs.put("profile.default_content_setting_values.notifications", 2);
28
29         // Create object of ChromeOption class
30         ChromeOptions options = new ChromeOptions();
31
32         // Set the experimental option
33         options.setExperimentalOption("prefs", prefs);
34     }
35
36     public void fetchdata(String url) {
37
38         driver.get(url);
39
40         WebElement element = driver.findElement(By.xpath("//div[@class='content']"));
41
42         String content = element.getText();
43
44         System.out.println(content);
45     }
46
47     public void fetchdata_justdial(String url) {
48
49         driver.get(url);
50
51         WebElement element = driver.findElement(By.xpath("//div[@class='content']"));
52
53         String content = element.getText();
54
55         System.out.println(content);
56     }
57
58     public void f_just(String url) {
59
60         driver.get(url);
61
62         WebElement element = driver.findElement(By.xpath("//div[@class='content']"));
63
64         String content = element.getText();
65
66         System.out.println(content);
67     }
68
69     public void best_h(String url) {
70
71         driver.get(url);
72
73         WebElement element = driver.findElement(By.xpath("//div[@class='content']"));
74
75         String content = element.getText();
76
77         System.out.println(content);
78     }
79
80     public static void main(String[] args) {
81
82         Mouthshut mouthshut = new Mouthshut();
83
84         mouthshut.fetchdata("https://www.justdial.com/Nashik/SNJB-s-Late-Sau-Kantabai-Bhavarlalji-Jain-College-Of-Engineering-Neminagar-Chandwad/0253PX253-X253-151019113056-B2P9_BZDET/reviews/page-6");
85
86     }
87 }
```

- Task List:** Shows the tasks for the current file.
- Outline:** Shows the class structure with methods **fetchdata**, **fetchdata_justdial**, **f_just**, **best_h**, and **main**.
- Console:** Shows the output: **<terminated> Mouthshut [Java Application] C:\Program Files\eclipse-java-photon-R-win32\eclipse\jre\bin\javaw.exe (Sep 30, 2018, 9:28:33 AM)**
- Bottom Bar:** Shows the operating system taskbar with icons for various applications.

After Execution Code Mouthshut.java Chrome Browser Opened Automatically with Specified website you can also see the output of rating in console as well as file you created.

```

SeleniumProject - ReviewProj/src/Mouthshut.java - Eclipse IDE
File Edit Source Refactor Search Project Run Window Help
Package Explorer Mouthshut.java
Ghanshyam_Selenium ReviewProj
JRE System Library [JavaSE-1.8]
src
(default package)
Mouthshut.java
Referenced Libraries selenium-server-standalone
Problems Declaration Console Coverage
<terminated> Mouthshut [Java Application] C:\Program Files\eclipse-jee-photon-R-win32\eclipse\jre\bin\javaw.exe (Sep 30, 2018, 9:28:33 AM)
Starting ChromeDriver 2.40.565498 (ead82db3280dd843ebfb08a625e3eb905cf9ab) on port 40918
Only local connections are allowed.
Sep 30, 2018 9:28:41 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
Name: Mr Ajay
Post :Good

Name: Mr Vaibhav
Post :Excellent

Name: Jayesh Jadhav
Post :Excellent

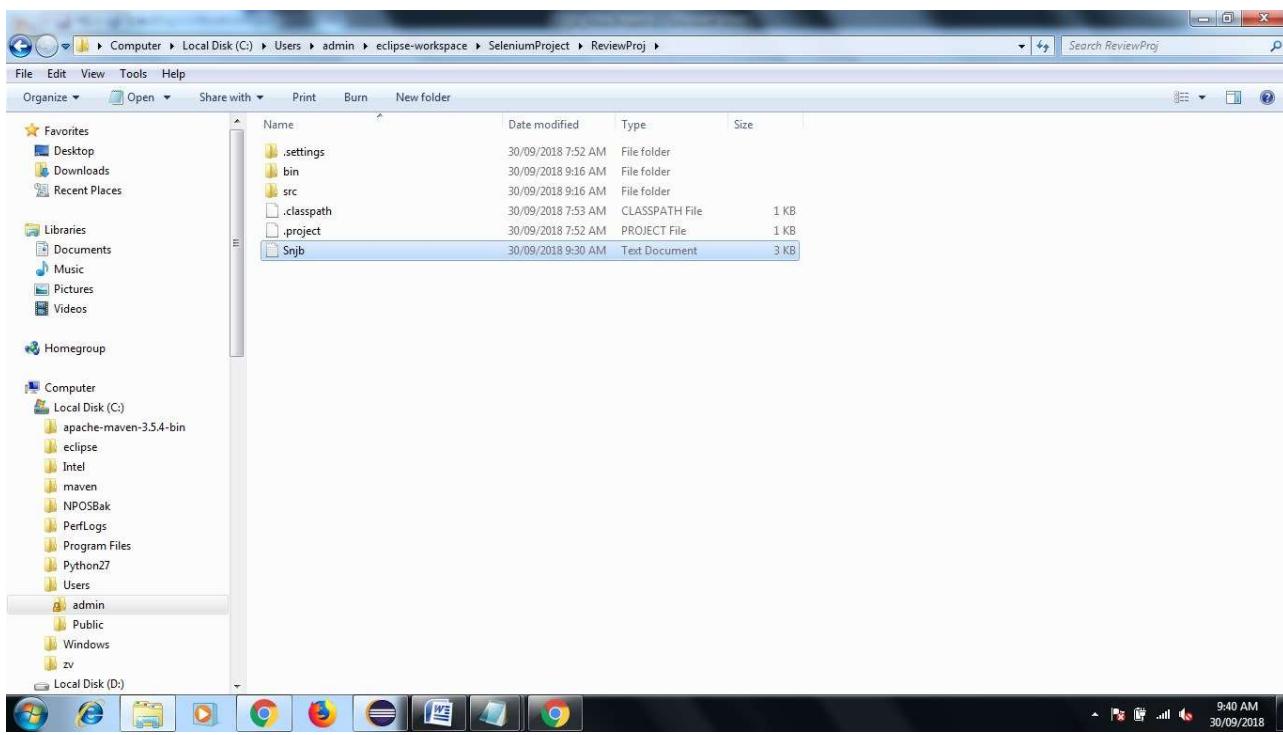
Name: Mr Yash
Post :Good

Name: Mr Prathmesh
Post :Very Good

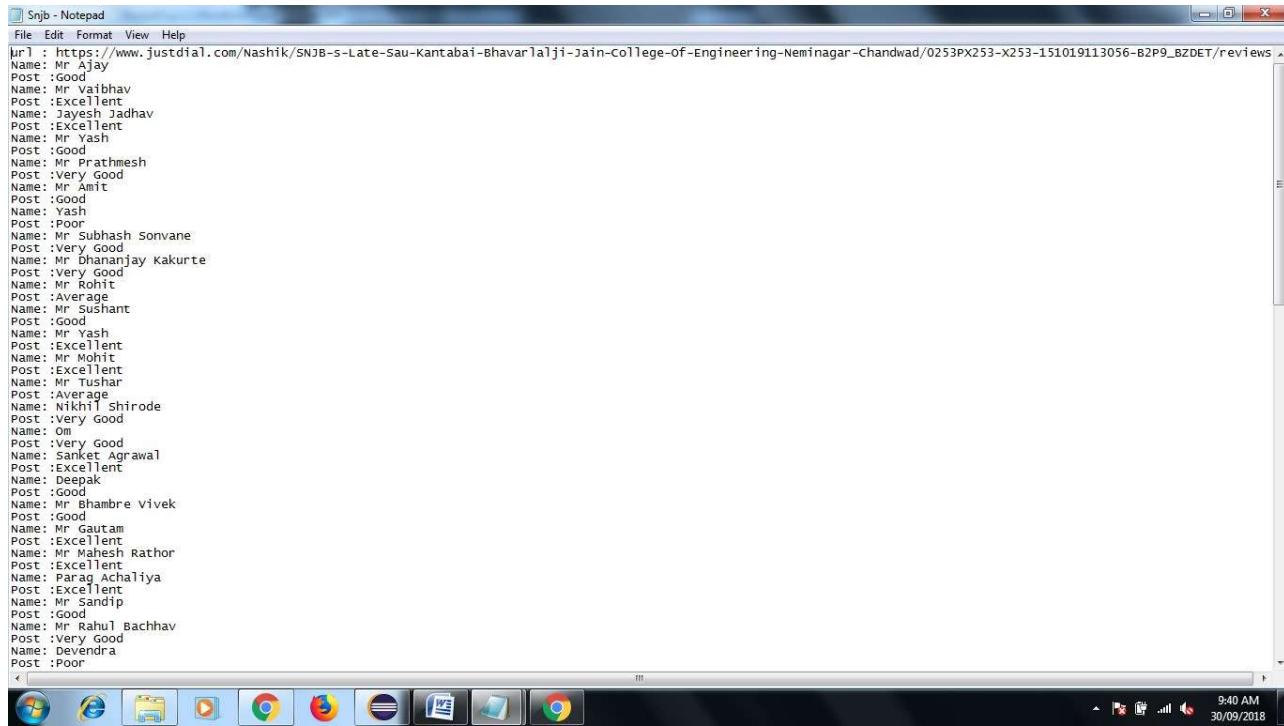
Name: Mr Amit

```

Now Lets Check the Feedback of All Customer go to your Eclipse Workspace Path my path is C:\Users\admin\eclipse-workspace\SeleniumProject\ReviewProj\Snjb.txt



After Open Txt File See the Output



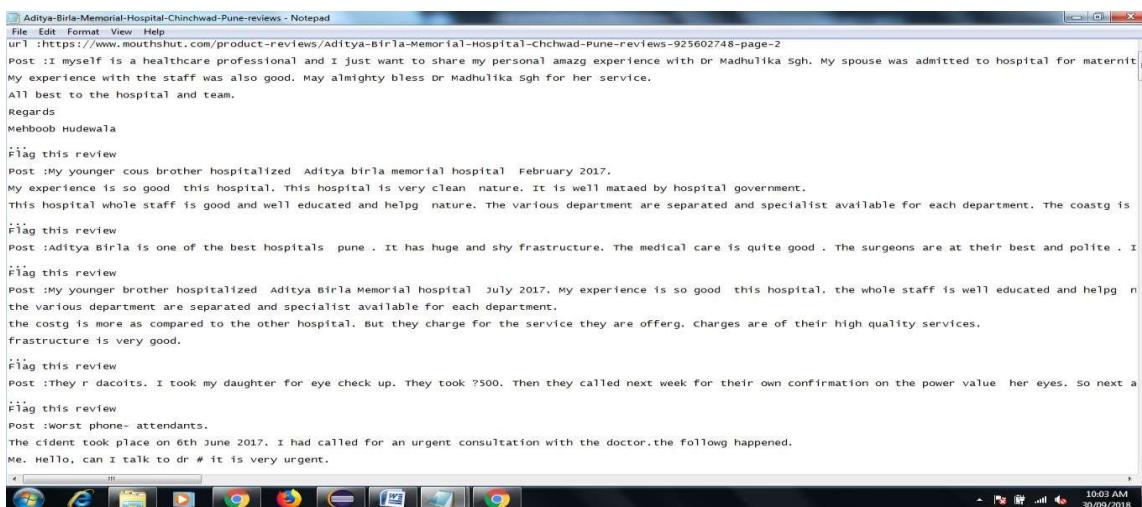
```
Snjb - Notepad
File Edit Format View Help
url : https://www.justdial.com/Nashik/SNJB-s-Late-Sau-Kantabai-Bhavarlalji-Jain-College-of-Engineering-Neminagar-Chandwad/0253PX253-x253-151019113056-B2P9_BZDET/reviews
Name: Mr Ajay
Post :Good
Name: Mr Subhash
Post :Excellent
Name: Jayesh Jadhav
Post :Excellent
Name: Mr Yash
Post :Good
Name: Mr Prathmesh
Post :Very Good
Name: Mr Amit
Post :Good
Name: Yash
Post :Poor
Name: Mr Subhash Sonvane
Post :Very Good
Name: Mr Dhananjay Kakurte
Post :Very Good
Name: Mr Rudit
Post :Average
Name: Mr Sushant
Post :Good
Name: Mr Yash
Post :Excellent
Name: Mr Motit
Post :Excellent
Name: Mr Tushar
Post :Average
Name: Nikhil shirode
Post :Very Good
Name: Om
Post :Very Good
Name: Sanjay Agrawal
Post :Excellent
Name: Deepak
Post :Good
Name: Mr Bhambre vivek
Post :Good
Name: Mr Gautam
Post :Excellent
Name: Mr Mahesh Rathor
Post :Excellent
Name: Parag Achaliya
Post :Excellent
Name: Mr Sandip
Post :Good
Name: Mr Rahul Bachhav
Post :Very Good
Name: Devenra a
Post :Poor
```

In This you can Collect all different kind of colleges, hospital rating and review from various website like mouthshout.com, Justdial.com via Selenium Web driver Tool

Module-3 Now I want Collect Review of Aditya-Birla-Memorial-Hospital-Chinchwad-Pune

Link- <https://www.mouthshut.com/product-reviews/Aditya-Birla-Memorial-Hospital-Chinchwad-Pune-reviews-925602748-page-2>

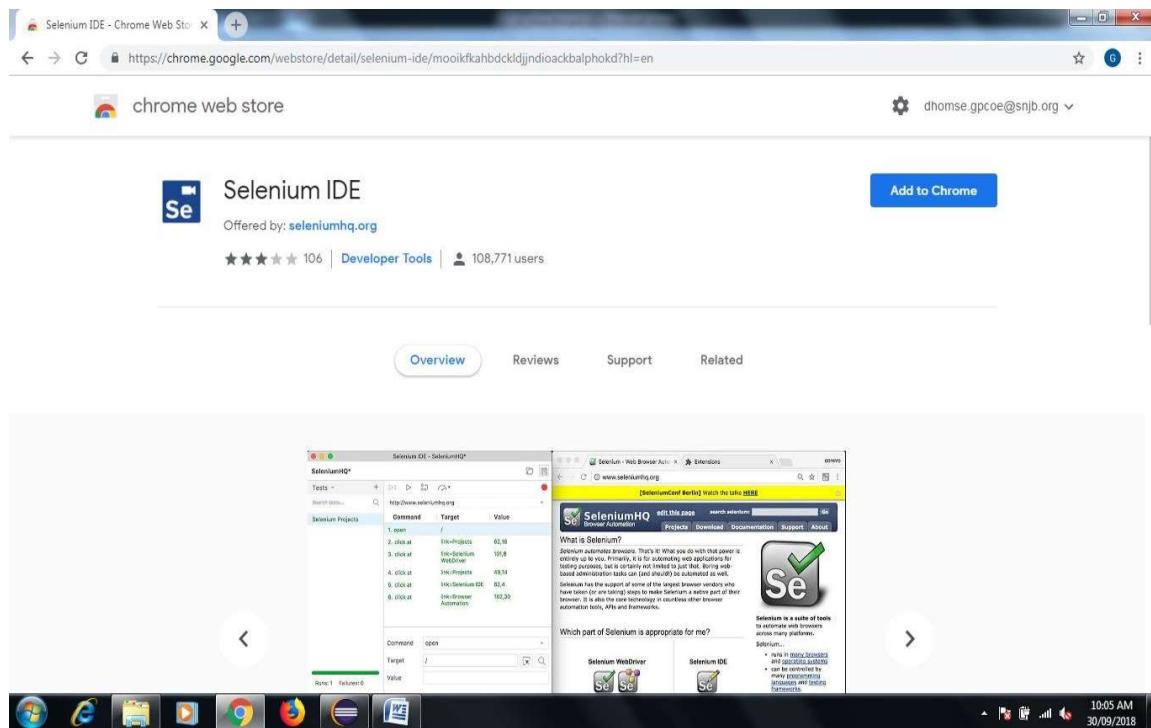
After Execution of Code-

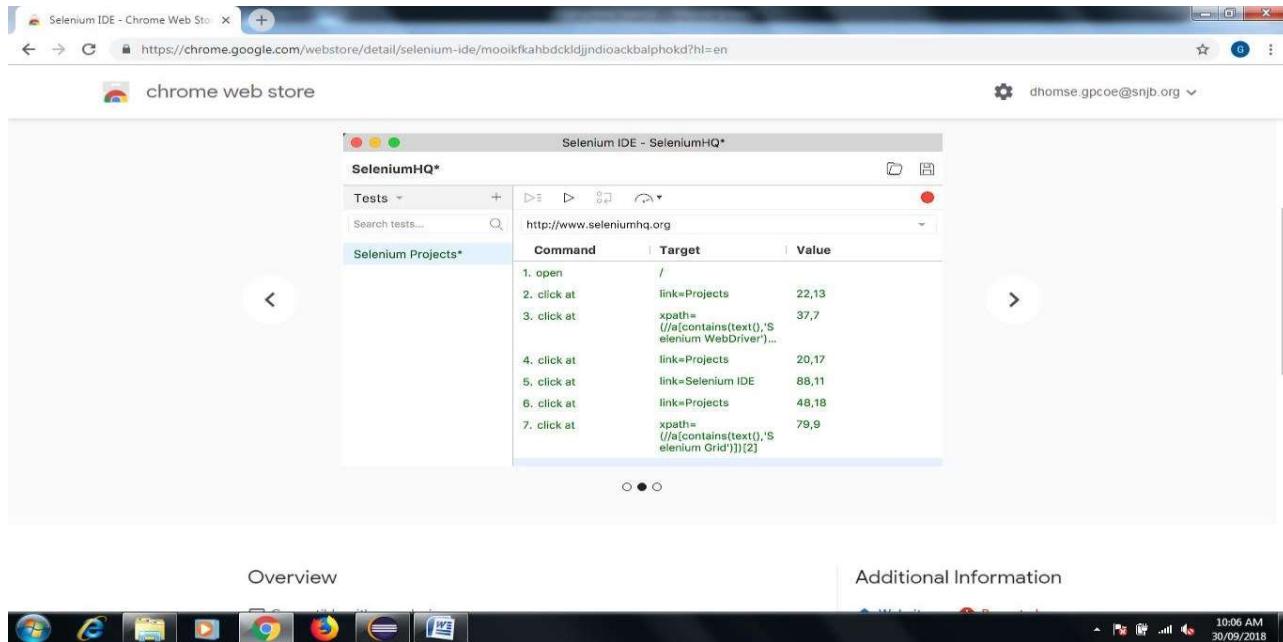


```
Aditya-Birla-Memorial-Hospital-Chinchwad-Pune-reviews - Notepad
File Edit Format View Help
url : https://www.mouthshut.com/product-reviews/Aditya-Birla-Memorial-Hospital-Chinchwad-Pune-reviews-925602748-page-2
Post : I myself is a healthcare professional and I just want to share my personal amazg experience with dr Madhulika sgh. My spouse was admitted to hospital for maternity. My experience with the staff was also good. May almighty bless Dr Madhulika sgh for her service.
All best to the hospital and team.
Regards
Mehboob Hudewala
Flag this review
Post :My younger cous brother hospitalized Aditya birla memorial hospital February 2017.
My experience is so good this hospital. This hospital is very clean nature. It is well mated by hospital government.
this hospital whole staff is good and well educated and helpg nature. The various department are separated and specialist available for each department. The coastig is
Flag this review
Post :Aditya Birla is one of the best hospitals pune . It has huge and shy frasrtucture. The medical care is quite good . The surgeons are at their best and polite . I
Flag this review
Post :My younger brother hospitalized Aditya Birla Memorial hospital July 2017. My experience is so good this hospital. the whole staff is well educated and helpg in
the various department are separated and specialist available for each department.
the costg is more as compared to the other hospital. But they charge for the service they are offerg. charges are of their high quality services.
frasrtucture is very good.
Flag this review
Post :They r dacoits. I took my daughter for eye check up. They took ?500. Then they called next week for their own confirmation on the power value her eyes. so next a
Flag this review
Post :worst phone- attendants.
The cident took place on 6th June 2017. I had called for an urgent consultation with the doctor.the followng happened.
Me. Hello, can I talk to dr # it is very urgent.
```

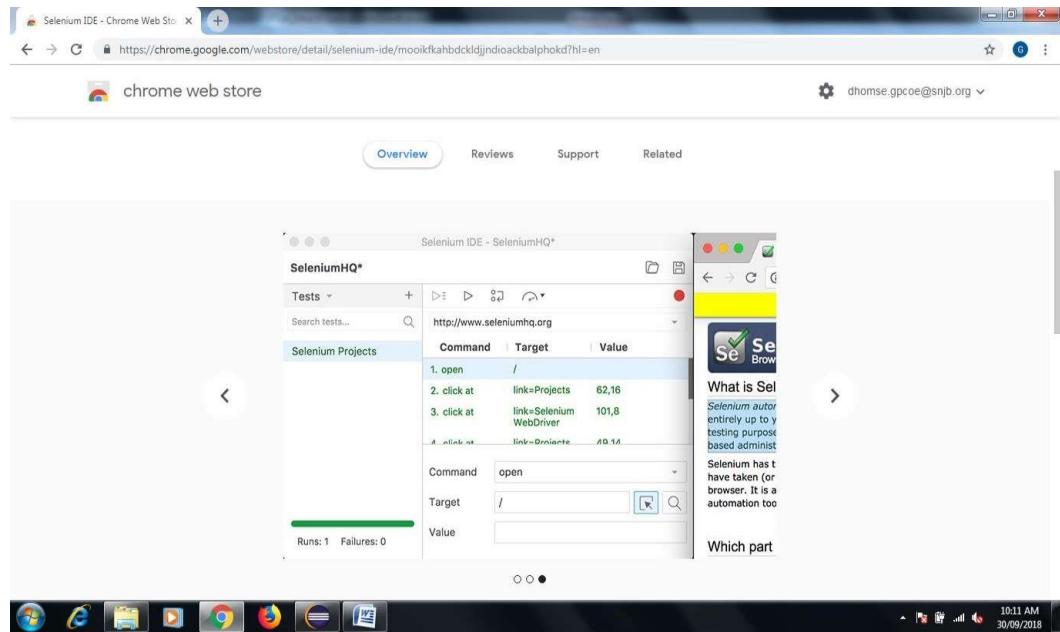
Selenium IDE:

1. Download Selenium IDE Chrome Extension from following Link
<https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldijndioackbalphokd?hl=en>
2. Click on Add to Chrome
3. Check the icon on of IDE Square in right side after browser after successfully installation to chrome
4. Now Click on record button do some operation on website...see the report in console of IDE command target and value....in this way you also find out xpath etc information.





You can also check the Automated Test Case Pass or Fails see the following figure



Conclusion:

In this way you learn how to use Selenium Open Source Tool for perform Automation Testing on web based application.