Experiment No. – 01

Title: - Write TEST Scenario for Gmail Login Page

Theory: -

Software testing is known as a process for validating and verifying the working of a software/application. It makes sure that the software is working without any errors, bugs, or any other issues and gives the expected output to the user. The software testing process doesn't limit to finding faults in the present software but also finding measures to upgrade the software in various factors such as efficiency, usability, and accuracy. So, to test software the software testing provides a particular format called a **Test Case**.

What is Test Case?

A test case is a defined format for software testing required to check if a particular application/software is working or not. A test case consists of a certain set of conditions that need to be checked to test an application or software i.e. in more simple terms when conditions are checked it checks if the resultant output meets with the expected output or not. A test case consists of various parameters such as Id, condition, steps, input, expected result, result, status, and remarks.

Parameters of a Test Case: -

- **Module Name:** Subject or title that defines the functionality of the test.
- **Test Case Id:** A unique identifier assigned to every single condition in a test case.
- **Tester Name:** The name of the person who would be carrying out the test.
- **Test scenario: The test** scenario provides a brief description to the tester, as in providing a small overview to know about what needs to be performed and the small features, and components of the test.
- **Test Case Description:** The condition required to be checked for a given software. for eg. Check if only numbers validation is working or not for an age input box.
- **Test Steps:** Steps to be performed for the checking of the condition.
- **Prerequisite:** The conditions required to be fulfilled before the start of the test process.
- **Test Priority:** As the name suggests gives the priority to the test cases as in which had to be performed first, or are more important and which could be performed later.
- **Test Data:** The inputs to be taken while checking for the conditions.
- **Test Expected Result:** The output which should be expected at the end of the test.
- **Test parameters:** Parameters assigned to a particular test case.
- **Actual Result:** The output that is displayed at the end.
- **Environment Information:** The environment in which the test is being performed, such as the operating system, security information, the software name, software version, etc.
- **Status:** The status of tests such as pass, fail, NA, etc.
- **Comments:** Remarks on the test regarding the test for the betterment of the software.

When do we Write Test Cases?

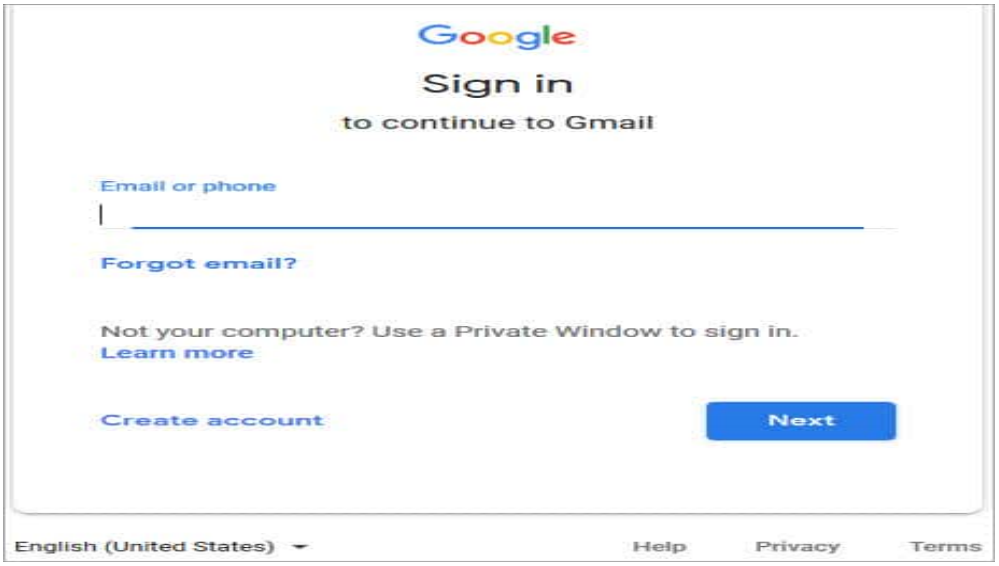
Test cases are written in different situations:

- **Before development:** Test cases could be written before the actual coding as that would help to identify the requirement of the product/software and carry out the test later when the product/software once gets developed.
- **After development:** Test cases are also written directly after coming up with a product/software or after developing the feature but before the launching of a product/software as needed to test the working of that particular feature.
- **During development:** Test cases are sometimes written during the development time, parallelly. so whenever a part of the module/software gets developed it gets tested as well.

Types of Test Cases

- **Functionality Test Case:** The functionality test case is to determine if the interface of the software works smoothly with the rest of the system and its users or not. Black box testing is used while checking for this test case, as we check everything externally and not internally for this test case.
- **Unit Test Case:** In unit test case is where the individual part or a single unit of the software is tested. Here each unit/ individual part is tested, and we create a different test case for each unit.
- **User Interface Test Case:** The UI test or user interface test is when every component of the UI that the user would come in contact with is tested. It is to test if the UI components requirement made by the user are fulfilled or not.
- **Integration Test Case: Integration** testing is when all the units of the software are combined together and then they are tested. It is to check that each component and its units work together without any issues.
- **Performance Test Case:** The performance test case helps to determine response time as well as the overall effectiveness of the system/software. It’s to see if the application will actually handle real-world expectations.
- **Database Test Case:** Also known as back-end testing or data testing checks that everything works fine with respect to the database. Testing cases for tables, schema, triggers, etc. are done.
- **Security Test Case:** The security test case helps to determine that the application restricts actions as well as permissions wherever necessary. Encryption and authentication are considered as main objectives with regard to the security test case. The security test case is done to protect and safeguard the data of the software.
- **Usability Test Case:** Also known as a user experience test case, it checks how user-friendly or easy to approach a software would be. Usability test cases are designed by the User experience team and performed by the testing team.
- **User Acceptance Test Case:** The user acceptance case is prepared by the testing team but the user/client does the testing and review if they work in the real-world environment.

Test Cases – Gmail Login



Sr. No	TestCase_ID	Testcase Objective
1	Login_01	To verify Login functionality with valid email id and valid password.
2	Login_02	To verify Login functionality with valid email id and invalid password.
3	Login_03	To verify Login functionality with invalid email id and valid password.
4	Login_04	To verify Login functionality with invalid email id and invalid password.
5	Login_05	To verify Login functionality with blank email id and valid password.
6	Login_06	To verify Login functionality with a valid email id and a blank password.
7	Login_07	To verify Login functionality with a blank email id and a blank password.
8	Login_08	To verify Login functionality with a valid phone number and valid password.
9	Login_09	To verify Login functionality with valid phone numbers and invalid passwords.
10	Login_10	To verify Login functionality with invalid phone number and valid password.
11	Login_11	To verify Login functionality with invalid phone numbers and invalid passwords.
12	Login_12	To verify Login functionality with blank phone number and valid password.

13	Login_13	To verify the length of the email id and password field.
15	Login_15	To verify that the error message displays when any field is left blank.
16	Login_16	To verify Tab key functionality on the Login page.
17	Login_17	To verify the alignment of each field of the login page.
18	Login_18	To verify that each mandatory field has an asterisk for validation.
19	Login_19	To verify that each field has a proper placeholder.
20	Login_20	To verify that entered multiple times incorrect password.
21	Login_21	To verify that show the password EYE icon on the Password text field.
22	Login_22	To verify that Welcome message after successful login.
23	Login_23	To verify that Forgot Email functionality.
24	Login_24	To verify that Create Account functionality.
25	Login_25	To verify that Next button with multiple times click events
26	Login_26	To verify that Forgot Password functionality.
27	Login_27	To verify if the Password text format is encrypted or not.
28	Login_28	To verify that back button functionality after log out.
29	Login_29	To verify that message for entered invalid inputs.
30	Login_30	To verify that Sign In button with a click and Enter key events.

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. – 02

Title: - TEST Scenario for Gmail Login Page

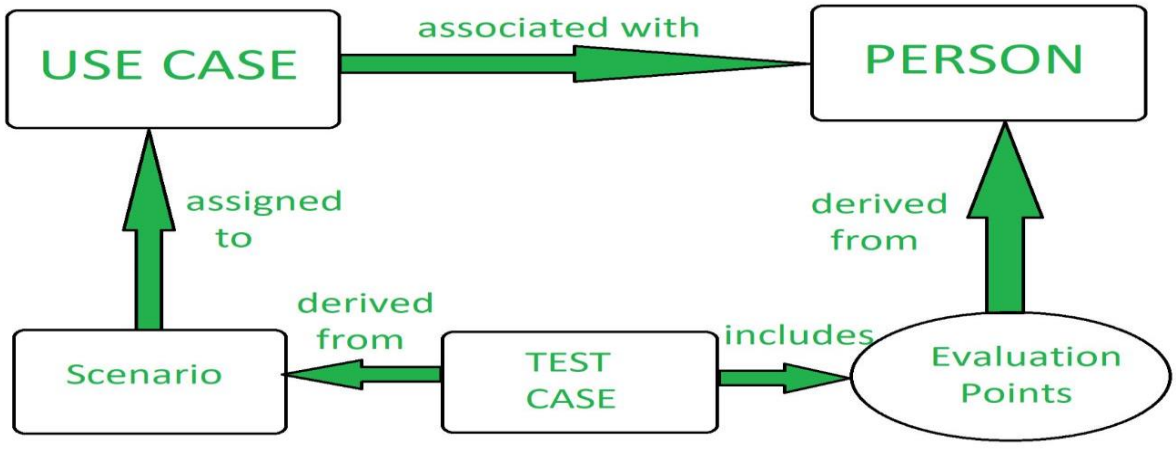
Theory: -

Scenario Testing is a Software_Testing_Technique that uses scenarios i.e., speculative stories to help the tester work through a complicated problem or test system. The ideal scenario test is a reliable, complicated, convincing or motivating story the outcome of which is easy to assess. Usually, these tests are different from test cases as the test cases are single steps whereas scenarios cover a number of steps. Scenario testing is performed to ensure that the end-to-end functioning of software and all the process flow of the software are working properly. In scenario testing, the testers assume themselves to be the end users and find the real-world scenarios or use cases which can be carried out on the software by the end user. In scenario testing, the testers take help from clients, stakeholders and developers to create test scenarios. Scenario testing helps testers to know how the software will exactly work when end user will use it. As the scenario testing tests the business process flow of the software so it helps in figure out a lot of defects which cannot be found with the help of other testing. Scenario testing is carried out by creating test scenarios which copy the end user's usage. A test scenario is a story which describes the usage of the software by an end user.

Characteristics of Scenario Testing: A scenario test has five key characteristics:

- Story
- Motivating
- Credible
- Complex
- Easy to evaluate

Scenario Testing Process:



Methods in Scenario Testing: There are two methods in scenario testing:

1. **System scenarios:** Scenario tests used in this method are only those sets of realistic, user activities that cover various components in the system.
2. **Use-case and role-based scenarios**: In the use-case and role-based scenario method the focus is specifically on how the system is used by a user with different roles and environment.

Risks of Scenario Testing:

- Scenario testing is complex involving many features.
- Scenario testing is not designed for coverage of the program.
- Scenario testing is often heavily documented and used time and again.

Test Scenario for Gmail – Inbox Functionality

Test Case for Gmail – Inbox Functionality

1. Verify that all the read and unread emails are displayed in the inbox
2. Verify that the recently received email or unread emails are highlighted in bold in the Inbox section.
3. Verify that the recently received email has correct sender ‘s name or email id, subject of the email, its preview and date or time.
4. Verify that the recently received email’s sender’s name or email id, subject of the email, and date or time should be in bold and preview text shouldn’t be in bold.
5. Verify that the attachment icon is displayed next to the preview text of the email, if the email has any attachment.
6. Verify that the Archive, Delete, Mark as read, Snooze options are displaying on hovering the unread email.
7. Verify that the Archive, Delete, Mark as unread, Snooze options are displaying on hovering the read email.
8. Verify that the Email id, add to contacts, Open detailed view, Send email, Send message, Start video call, Schedule event options are displaying when we hover on the name/email of the read/unread email.
9. Verify that the user is navigated to the email content when clicking on the email in the inbox.
10. Verify that the content of the email is displayed correctly without any formatting issues.
11. Verify that the attachment in the email is downloadable or not.
12. Verify that the attachments can be downloaded as a single zip file.
13. Verify that the attachments can be downloaded individually.
14. Verify that the attachments can be viewable in the browser itself without downloading.
15. Verify that the attachment is downloading in zip format, if the attachment size is more than 1 MB.
16. Verify that the attachments are scanned for viruses once we try to download the file.
17. Verify that the Reply and Forward buttons are displaying in the bottom of the email content.
18. Verify that all the read emails are not highlighted.
19. Verify that unread emails count is displayed beside ‘Inbox’ text in the left sidebar of Gmail.
20. Verify that unread emails count is increased as per the number of new emails we received.
21. Verify that the unread emails count is increased when we mark an email as unread.
22. Verify that the unread emails count is decreased when we mark an email as read or opened.
23. Verify that email recipients in CC are visible to all the users whose emails are present.
24. Verify that email recipients in BCC are not visible to other users in the TO, CC or BCC section.

25. Verify that email can be received from other domains like Hotmail, Outlook, Yahoo mail or any other company domains.

Test Cases for Gmail – Compose Mail Functionality

1. Verify that the 'New Message' popup is displaying on clicking on the Compose email button.
2. Verify that the email ids can be entered in the fields To, CC, and BCC
3. Verify that the auto suggestion is working properly based on the existing contacts while typing email ids in the To, CC, and BCC fields.
4. Verify that multiple commas separated email ids can be entered in the To, CC, and BCC sections
5. Verify that the text can be entered in the Subject text box.
6. Verify that the text can be entered in the email body area.
7. Verify that the editor options such as font-family, font-size, bold, italic, underline, text color, etc., are working and allowing user to format the email body text.
8. Verify that the user can add files in the attachment section.
9. Verify that the user can add images in the email body area.
10. Verify that the email is delivered to all the intended users in the To, CC, BCC.
11. Verify that the sent emails are available in the Sent Mail section
12. Verify that the email is delivered to all the intended users who are using non-Gmail ids such as Hotmail, Outlook, Yahoo mail etc.,
13. Verify that the emails composed and not sent are stored in the Draft section.
14. Verify that the maximum number of emails can be entered in the To, CC, and BCC fields.
15. Verify that the maximum number of characters allowed in the subject text box.
16. Verify that the maximum number of characters allowed in the email body text filed.
17. Verify that the maximum number of attachments and size of the attachment allowed in an email.
18. Verify whether a warning popup message is displayed when we try to send an email without Subject line.
19. Verify whether an error popup message is displayed when we enter an incorrect text entered in the email To, CC, BCC fields.
20. Verify whether the Schedule send functionality is working properly or not
21. Verify whether the signature added in the settings is displayed or not when we compose an email.

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.

Experiment No. – 03

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

Theory: -

What is WhatsApp?

WhatsApp is one of the most used instant messaging apps in the world. It's well-known for its features, like end-to-end encryption, which makes it very secure. But WhatsApp is also a great test case for any mobile app or software company seeking regulatory approval. This article covers various WhatsApp test scenarios like test cases on WhatsApp Group & WhatsApp test scenarios Profile pictures.

Test Cases For WhatsApp

Below we are going to mention all the possible integration testing scenarios for WhatsApp, which have also covered positive and negative test cases for WhatsApp or negative test scenarios for WhatsApp:

- Installation Testing
- Usability Testing
- Performance Testing
- WhatsApp Group Feature
- Admin Feature Of WhatsApp Group
- Adding Feature Of WhatsApp Group
- Delete Feature Of WhatsApp Group
- Contact
- Text Field
- WhatsApp Status
- WhatsApp New broadcast
- WhatsApp Web
- WhatsApp Settings
- Profile Picture
- Individual Chats
- Group Chats
- Audio Calls

- Video Calls

Installation Testing Test Scenarios For WhatsApp

- Check if the user can download the WhatsApp app from the Play Store or App Store or not.
- Check whether the successful installation is on mobile.
- Check if the user can register with a new phone number.
- Check whether the user can get a verification code on his mobile.
- Check the maximum number of incorrect attempts of verification code.
- Check one phone number should be registered on only one WhatsApp.
- Check if successful registration on mobile.
- Check if contacts are imported to the WhatsApp contact list or not.
- Check if the user can upload a profile picture or not.

Usability Testing WhatsApp Test Scenarios

- Check if WhatsApp is easy to use or not.
- Check whether the Search, Chat, Status, Contacts, and Calls sections are available on-screen.
- Check if the user can see all sections individually.
- Check if the user can see unread messages contacts and group-wise.

Test Scenarios For Performance Testing

- Check the WhatsApp performance when the application is continuously used.
- Check the WhatsApp performance when users use multiple chats at the same time.
- Check the WhatsApp performance when the user uses multiple functions like sending videos, images, and text simultaneously.

Test Cases For WhatsApp Group Features

WhatsApp Test Cases

Test Scenarios For Admin Feature Of WhatsApp Group

- Check if an admin can add others as Admin.
- Check admin can remove it from the group.
- Check admin can add users to the group.
- Check admin can restrict users.
- Check admin can remove others from admin.

Test Scenarios For Adding Feature Of Whatsapp Group

- Check if the admin can add people.
- Check if the admin can add 250 people to a group.
- Check the admin user able to add people with the invite link
- Check the admin can delete people and add them back to the group.

Test Scenarios For Delete Feature Of Whatsapp Group

- Check if the admin user can able to delete people.
- Check the admin user able to delete all people in the group
- Check the admin user can able to ban users.

Test Case For WhatsApp Contact

- Check the contact details shows the name and profile photos of the contacts.
- Check the status of the contacts will be displayed like the status of contact as Online or last seen with date, time Ex: last seen 10:35 PM, Last Seen 25/12/2014 05:04 AM.
- Check on the click of the back button; the user should be redirected to the chat list.

Test Case For WhatsApp Text Field

- Check the maximum length of the text field.
- Check the minimum length of the text field.
- Check the total number of characters the text field can allow.
- Check if the textbox accepts numbers only.
- Check the textbox allows decimal numbers.
- Check if the text field accepts alphabets.
- Verify the field accepts Upper case letters.
- Check the field accepts lowercase.
- Check the text field accepts both upper and lowercase alphabets.
- Check the text field accepts special characters.
- Check the field to accept HTML characters. (For Example <b>)
- Check the field accepts copy-paste functionality.
- Check if the cursor is displaying when you type.
- Check the text field allowed spaces and spaces between two letters.

Test Cases For WhatsApp Status

- Check whether the user can set DP or not.
- Check whether the user can update the DP or not.
- Check whether the user can set the status or not.
- Check whether the user can update the status or not.
- Check the Status of Privacy functionality.
- Check the user can see the status privacy options on the screen.
- Check the status privacy functionality with My Contacts options

- Check the status privacy functionality with My Contacts except for options.
- Check the status privacy functionality with the Only share with options
- Check whether the recent updates are displayed or not.
- Verify the user can see the number of unseen status notifications in the WhatsApp Status.

Test Scenarios For WhatsApp New broadcast

- Check the user can select contacts for the broadcast messages.
- Check whether the user can send and receive messages from broadcast or not.
- Check the user can send and receive messages from broadcast in an individual chat box.
- Check the maximum number of contacts for broadcast messages.

Test Scenarios For WhatsApp Web

- Check whether the user can see the Scan code screen on the mobile phone.
- Check whether the QR code is scanned from WhatsApp web or not.
- Check whether the user can get a message for another WhatsApp session with details.
- Check whether the user can log out from all WhatsApp web or not.

Test Scenarios For WhatsApp Settings

- Check whether the user can see setting options on the screen or not.
- Check the Account functionality with the Privacy option.
- Check the Account functionality with the Security option.
- Check the Account functionality with Two steps verification option.
- Check the Account functionality with the Change number option.
- Check the Account functionality with the Request account info option.
- Check the Account functionality with the Delete my account option.
- Verify the user can change the Chat setting with App language, Font size, Wall Paper, Chat back up, and Chat history.
- Check the user can see and change the notification settings.
- Check the user can see and change the Data and storage usage settings.

- Check if the How Help option is helpful for the users.

Test Cases For Whatsapp Group Chat

- Check if the admin can change information/group name.
- Check if the admin can change the group image.
- Check every user can share information.
- Check admin can restrict people from sharing information.
- Users can share different media on WhatsApp, like photos, videos, documents, links, and simple text.
- Check if the person is removed from the group; then, that user will cant see the updates.
- If an individual user has posted something in a group, then that individual user will be able to delete the information on the group.
- Check the individual can see the text status read/seen by other group users.

Test Cases for WhatsApp Profile Picture

- Check different options are available by clicking on the profile picture (Remove photo, Camera, gallery).
- Check if the remove confirmation message displays when you click the remove photo option.
- Check whether the user is redirected to the phone gallery after clicking on the gallery option.
- When the user clicks on the cancel button, it redirects the user to in gallery or not.
- Check whether the user selects the camera option and whether or not the mobile phone camera gets on.
- Check when the user clicks a photo from the camera and clicks on the true symbol page to redirect to the image resizes page or not.
- Check the user can select an image from the phone gallery.
- Check image resizes options are displaying after selecting an image from the gallery or taking pictures.

- Check user can resize the image or not.
- Check on click of a done button, the selected images should be displaying as DP or not and whether the user is getting a success message.

Test Scenarios For WhatsApp Individual Chats

- Check the Chat window that contains the entire chat list.
- Check the Chat window displays the contact numbers whose numbers are not saved on mobile.
- Check the Chat window displayed with all contacts with DP or without DP
- Check the Chat window is displayed on the group chat list.
- Check the Chat window displays the last updated chatting time.
- Check the Chat window displays the name of all contacts on the chat window.
- Check to click on one Chat contact, and then a new window should open with history.
- Check the user can see all delivered and received messages.
- Check the user can see the read or send time of messages.
- Check the user can send and receive text messages in the individual chatbox.
- Check the user can send and receive documents in the individual chatbox.
- Check the user can send and receive photos in an individual chatbox.
- Check the user can send and receive videos in an individual chatbox.
- Check the user can send and receive audio in an individual chat box.
- Verify the user can send and receive emotional icons in the individual chat boxes.
- Check the user can send and receive Contacts in the individual chat boxes.
- Check the user can send and receive Location in the individual chatbox.
- Check the user can send and receive GIFs in the individual chat boxes.
- Check the user can send and receive Stickers in the individual chatboxes.
- Verify the user can delete text, video, audio, locations, and documents in the individual chatboxes.
- Check the user can send recorded voice mail in an individual chatbox.
- Check the user can delete the entire chat history in the individual chatbox.

- Check the user can see contact details in the individual chat box.
- Verify the user can share images, links, and documents from media in the individual chatboxes.
- Verify the user can search specific chat history using the search option in the individual chatbox.
- Check the user can video call in the individual chat box.
- Check the user can voice call in the individual chat box.
- Check the user can mute the individuals in the individual chat boxes.
- Check the user can change the wallpaper.
- Check the users have options like Report, Block, Clear Chat, Export Chat, and Add Shortcut.

Test Scenarios For WhatsApp Group Chats

- Check whether the user can create a new one or not.
- Check the user can add multiple contacts from the contact list.
- Verify the user can insert the group name and select an image for DP.
- Check the user can add and remove contacts from the group.
- Check the user is able to delete a group.
- Check the user can send and receive text messages in the group.
- Check the user can send and receive documents in the group chat box.
- Check the user can send and receive photos in the group chat box.
- Check the user can send and receive videos in the group chat box.
- Check the user can send and receive audio in the group chat box.
- Check the user can send and receive emotions icons in the group chat box.
- Check the user can send and receive Contacts in the group chat box.
- Check the user can send and receive Location in the group chat box.
- Check the user can send and receive GIFs in the group chat box.
- Check the user can send and receive Stickers in the group chat box.
- Check the user can delete text, video, audio, locations, and documents in the group chat box.
- Check the user can send recorded voice mail in the group chat box.

- Check the user is able to make multiple video call in the group chat box.
- Verify the user can see the group contact information from Group Info in the group chat box.
- Check the user is able to share images, links, and documents from Group Media in the group chat box.
- Check the user is able to search specific chat history using the search option in the group chat box.
- Check the user is able to mute the group in the group chat box.
- Check the users have options like Report, Block, Clear Chat, Export Chat, and Add Shortcut.

Test Scenarios For WhatsApp Audio Calls

- Check the call history of audio is available or not.
- Check the call history is displayed with the date and time.
- Check the call history is displayed with updated time.
- Check whether the search functionality is working properly or not.
- Check whether the call log is removed from the call history or not.
- Check whether the call log is blocked from the call history or not.
- Check the new call log is working for the new call.
- Check the user is able to call or receive WhatsApp calls from the contact list.

Test Scenarios For WhatsApp Video Calls

- Check the call history of videos is available or not.
- Check the call history is displayed with the date and time.
- Check the call history is displayed with updated time.
- Check whether the search functionality is working properly or not.
- Check whether the video call log is removed from the call history or not.
- Check whether the video call log is blocked from the call history or not.
- Check the new video call log is working for the new video call.
- Check the user is able to call or receive WhatsApp video calls from the contact list.

Negative Test Cases For WhatsApp

- Check while hiding the last seen to other users it should not display.
- Check the blue ticks should not display when sending the messages.
- Check if the date time format is displaying in the wrong format.
- Check the displaying date and time in the wrong format.
- Try to send more than 10 images at a time. (as it allows only 10 images at a time)
- Try to send a video having a size > than the limited size.
- Copy and paste thousands of msgs at a time and observe the behavior.
- Send multiple messages to multiple users and groups.
- Try to send all emoticons at a time in one msg.

Conclusion

These are some of the test cases that you can use for the text field. There are many types of text inputs so you have to test according to the specification.

Experiment No. – 04

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

Theory: -

Defect:
A defect in a software product is also known as a bug, error or fault which makes the software produce an unexpected result as per the software requirements. For example; incorrect data, system hangs, unexpected errors, missing or incorrect requirements.

Defect Report:
A defect report is a document that has concise details about what defects are identified, what action steps make the defects show up, and what are the expected results instead of the application showing error (defect) while taking particular step by step actions.
Defect reports are usually created by the Quality Assurance team and also by the end-users (customers). Often customers detect more defects and report them to the support team of the software development since the majority of the customers curiously tries out every feature in the application. Now, you know what actually defect and defect reports are.

The reason behind why defect reports are created is to help developers to find out the defects easily and fix them up. A defect report is usually assigned by QA to a developer who then reads the report and reproduces the defects on the software product by following the action steps mentioned in the report. After that, the developer fixes the defects in order to get the desired outcome specified in the report.

That is why the defect reports are important and created carefully. Defect reports should be short, organized, straight to the point and covers all the information that the developer needs to detect the actual defects in the report by doing what and how the one written the defect report detected the defects.

It is usual for QA teams to get defect reports from the clients that are either too short to reproduce and rectify or too long to understand what actually went wrong.

For example,
Defect Description: The application doesn't work as expected.
Now, how in the world does a developer or QA know what went wrong which doesn't meet the client expectation?

In such a case, the developer report to the QA that he couldn't find any problem or he may have fixed any other error but not the actual one client detected. So that's why it's really important to create a concise defect report to get bugs fixed.

All right. You have a pretty good idea about what, whys and how's of a defect report. So it's time for what is inside the report.

A typical defect report contains the information in an xls Sheet as follows.

1. Defect ID:
Nothing but a serial number of defects in the report.
2. Defect Description:
A short and clear description of the defect detected.

3. Action Steps:
What the client or QA did in an application that results in the defect. Step by step actions they took.
4. Expected Result:
What results are expected as per the requirements when performing the action steps mentioned.
5. Actual Result:
What results are actually showing up when performing the action steps.
6. Severity:
Trivial (A small bug that doesn't affect the software product usage).

1. **Low –**
   A small bug that needs to be fixed and again it's not going to affect the performance of the software.
2. **Medium –**
   This bug does affect the performance. Such as being an obstacle to do a certain action. Yet there is another way to do the same thing.
3. **High –**
   It highly impacts the software though there is a way around to successfully do what the bug cease to do.
4. **Critical –**
   These bugs heavily impacts the performance of the application. Like crashing the system, freezes the system or requires the system to restart for working properly.

7. Attachments:
A sequence of screenshots of performing the step by step actions and getting the unexpected result. One can also attach a short screen recording of performing the steps and encountering defects. Short videos help developers and/or QA to understand the bugs easily and quickly.
8. Additional information:
The platform you used, operating system and version. And other information which describes the defects in detail for assisting the developer understand the problem and fixing the code for getting desired results.

Examples: -

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

Sample bug report 2: Application product Bug report sample
Application testing scenario:
Let's 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.

Conclusion: -

We successfully explored the concept of Create Defect Report for Any application or web application.

Experiment No. – 05

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

Theory: -

Selenium WebDriver- Installation

Selenium WebDriver installation process is completed in four basic steps:

1. Download and Install Java 8 or higher version.
2. Download and configure Eclipse or any Java IDE of your choice.
3. Download Selenium WebDriver Java Client
4. Configure Selenium WebDriver

1. Download and Install Java

We assume that you have already installed Java 8 or above on your machine and successfully configured the environment variables required to run and compile java programs.

Note: you'll need to have Java 8 installed to use Selenium 3.

However, you can download the latest version of Java Development Kit (JDK) from the link given below.

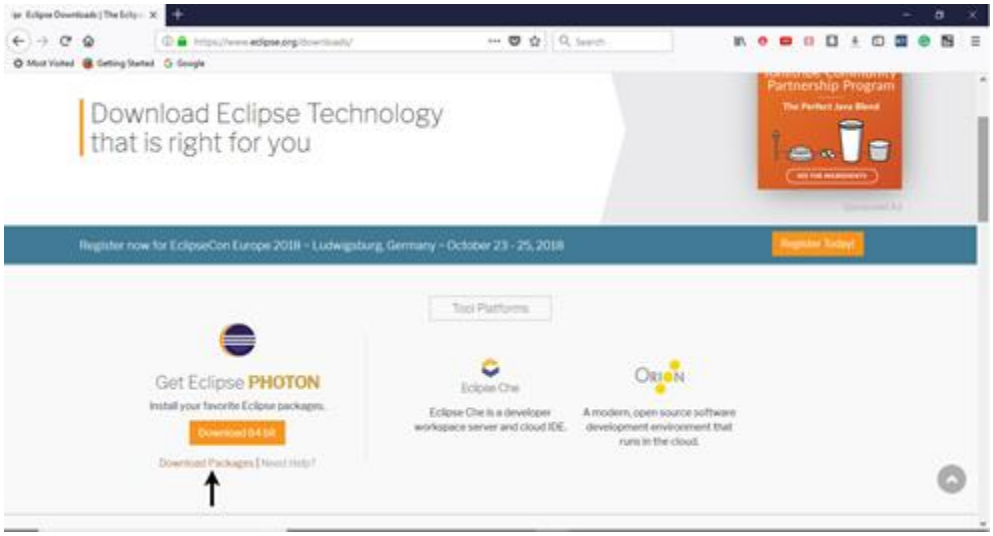
http://www.oracle.com/technetwork/java/javase/downloads/index.html

Once you have downloaded and installed the latest version of Java, you need to set path or configure the environment variables in your system. Refer the link given below to understand how we can set path and configure environment variables in Java.

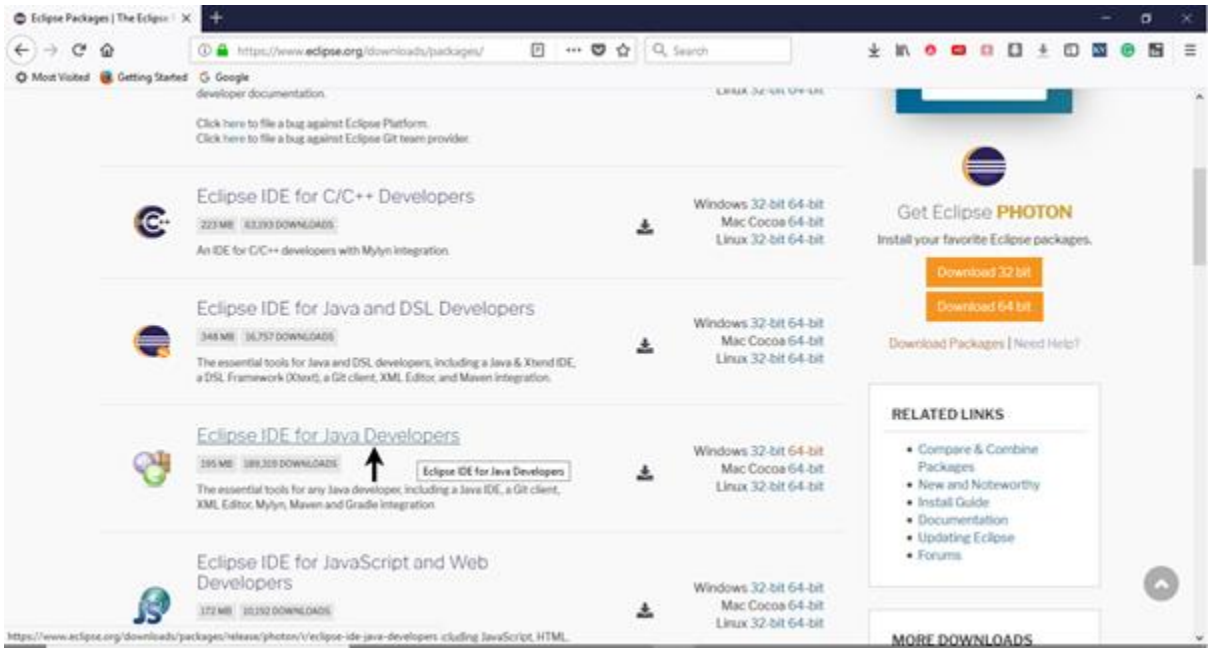
https://www.javatpoint.com/how-to-set-path-in-java

2. Download and Configure Eclipse IDE

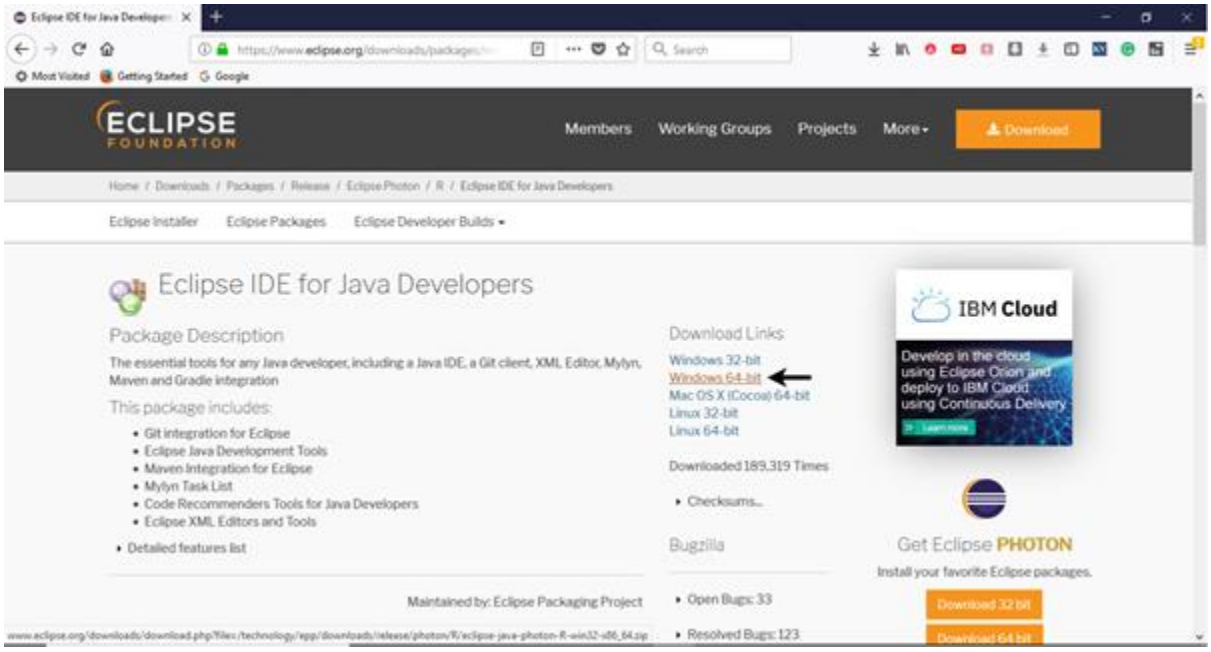
- Open URL: https://www.eclipse.org/downloads/ .
- Click on the "Download Packages" link (you can also download the IDE directly from the "downloads page" of Eclipse official website, but we will recommend you to navigate through the download packages section and get "Eclipse IDE for Java Developers").



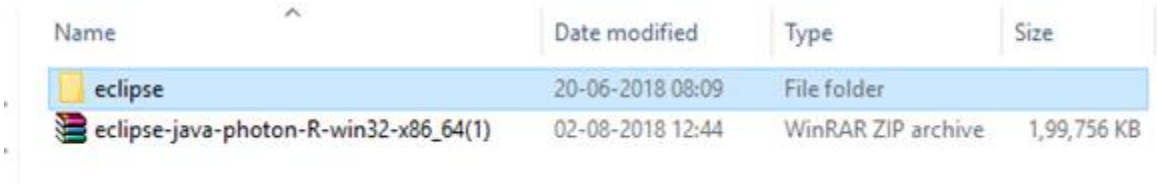
- It will redirect you to the "Download Packages" section. Scroll down through the webpage and click on "Eclipse IDE for Java Developers".



- Go to the Download Links section and click on "Windows 64-bit". You can also select other options to download based on the operating system you are currently working on.

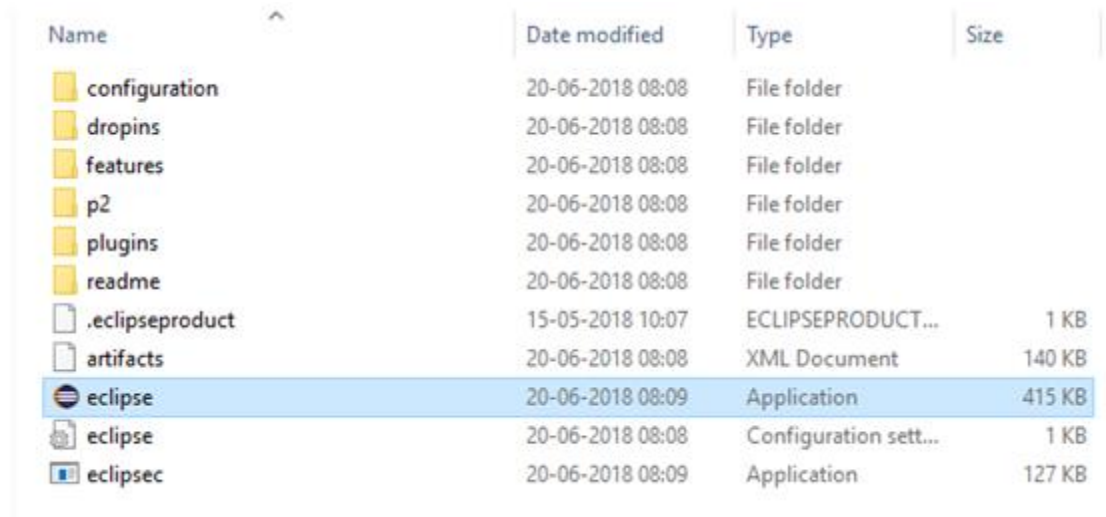


- The downloaded file would be in zipped format. Unpack the contents in a convenient directory.



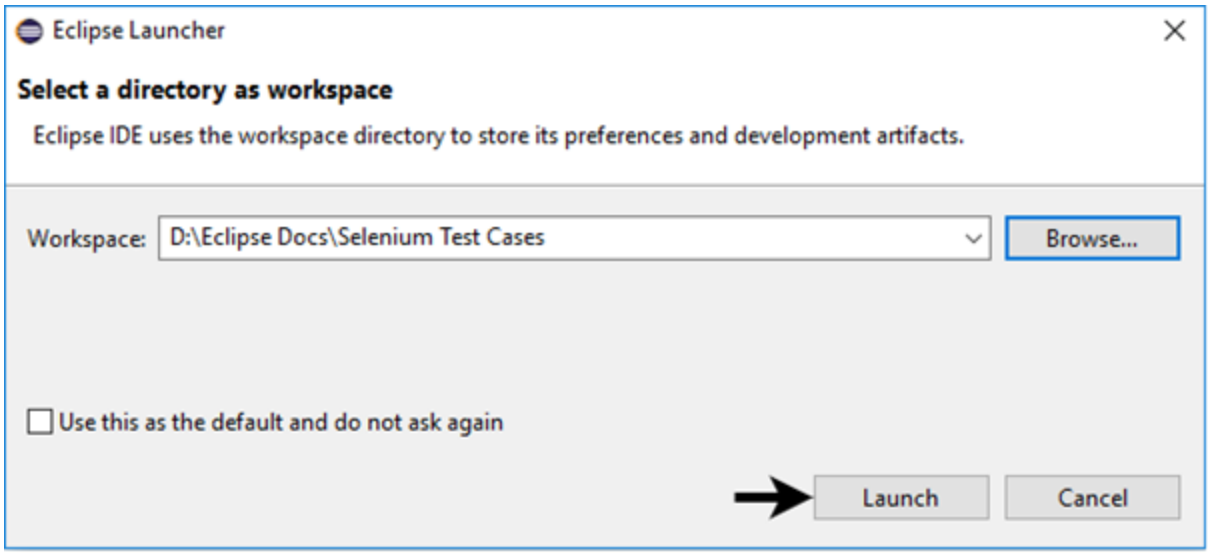
Name	Date modified	Type	Size
eclipse	20-06-2018 08:09	File folder	
eclipse-java-photon-R-win32-x86_64(1)	02-08-2018 12:44	WinRAR ZIP archive	1,99,756 KB

- Double click on "eclipse" (.exe file).

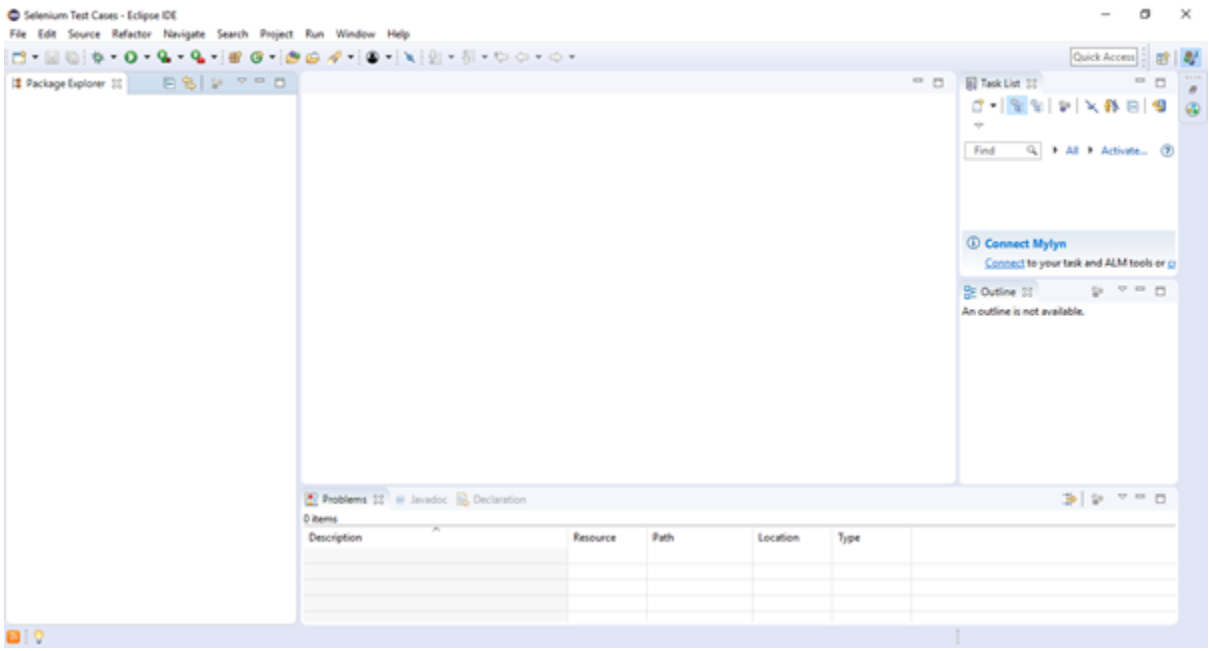


Name	Date modified	Type	Size
configuration	20-06-2018 08:08	File folder	
dropins	20-06-2018 08:08	File folder	
features	20-06-2018 08:08	File folder	
p2	20-06-2018 08:08	File folder	
plugins	20-06-2018 08:08	File folder	
readme	20-06-2018 08:08	File folder	
.eclipseproduct	15-05-2018 10:07	ECLIPSEPRODUCT...	1 KB
artifacts	20-06-2018 08:08	XML Document	140 KB
eclipse	20-06-2018 08:09	Application	415 KB
eclipse	20-06-2018 08:08	Configuration sett...	1 KB
eclipsec	20-06-2018 08:09	Application	127 KB

- To configure the workspace, select a convenient directory where you want to keep all of your Selenium trails and click on Launch button.



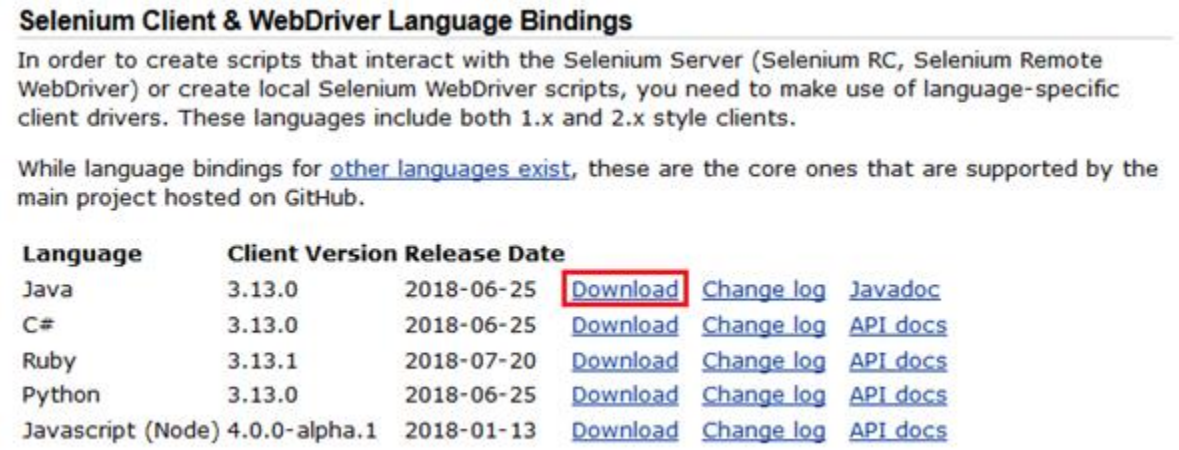
- It will launch the default interface of Eclipse IDE.



3. Download Selenium WebDriver Java Client

- Open URL: https://docs.seleniumhq.org/download/
  It will redirect you to the "downloads page" of Selenium official website.
- Scroll down through the web page and locate **Selenium Client & WebDriver Language Bindings**.

- Click on the "Download" link of Java Client Driver as shown in the image given below.



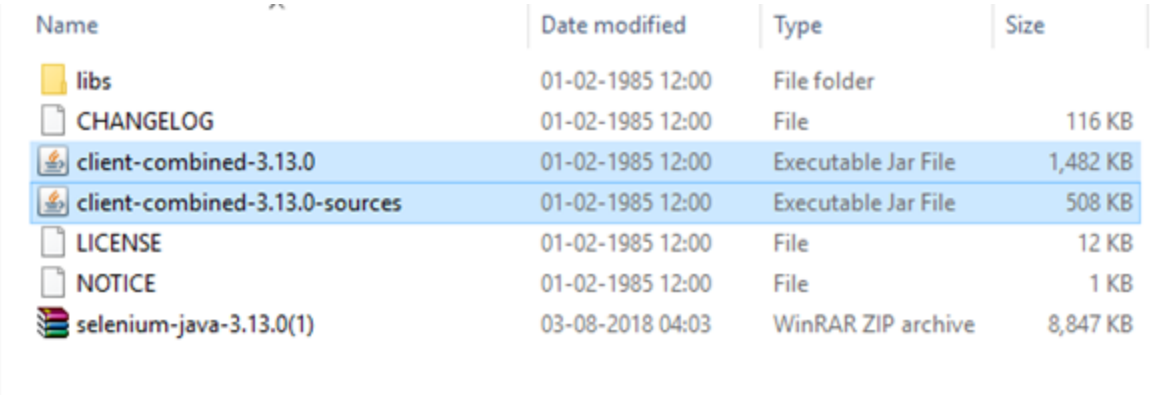
Selenium Client & WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Selenium RC, Selenium Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers. These languages include both 1.x and 2.x style clients.

While language bindings for other languages exist, these are the core ones that are supported by the main project hosted on GitHub.

Language	Client Version	Release Date			
Java	3.13.0	2018-06-25	Download	Change log	Javadoc
C#	3.13.0	2018-06-25	Download	Change log	API docs
Ruby	3.13.1	2018-07-20	Download	Change log	API docs
Python	3.13.0	2018-06-25	Download	Change log	API docs
Javascript (Node)	4.0.0-alpha.1	2018-01-13	Download	Change log	API docs

- The downloaded file would be in zipped format. Unpack the contents in a convenient directory. It contains the essential jar files required to configure Selenium WebDriver in Eclipse IDE.

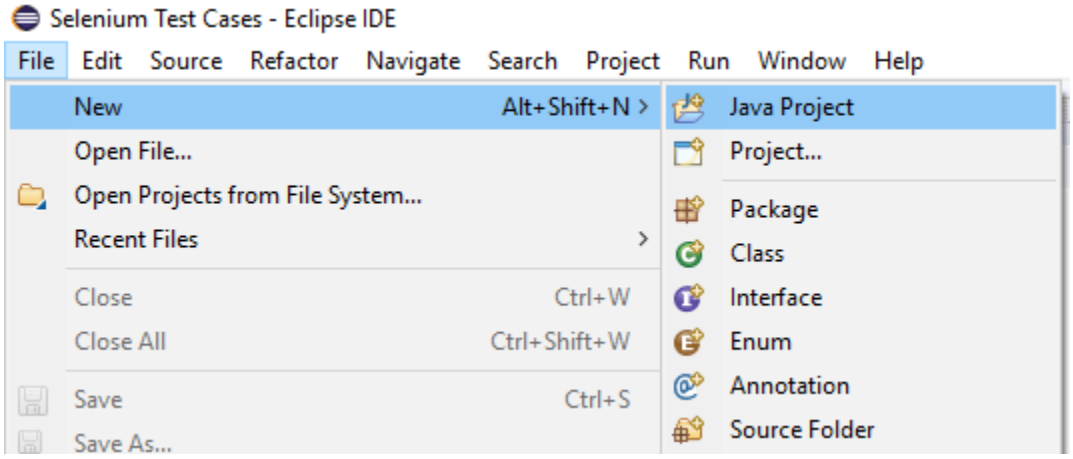


Name	Date modified	Type	Size
libs	01-02-1985 12:00	File folder	
CHANGELOG	01-02-1985 12:00	File	116 KB
client-combined-3.13.0	01-02-1985 12:00	Executable Jar File	1,482 KB
client-combined-3.13.0-sources	01-02-1985 12:00	Executable Jar File	508 KB
LICENSE	01-02-1985 12:00	File	12 KB
NOTICE	01-02-1985 12:00	File	1 KB
selenium-java-3.13.0(1)	03-08-2018 04:03	WinRAR ZIP archive	8,847 KB

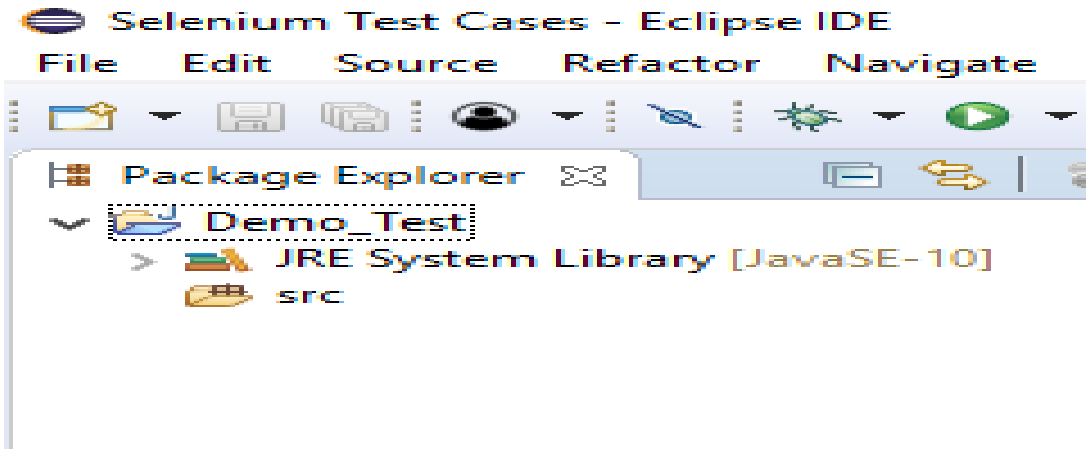
4. Configure Selenium WebDriver

Now we will configure our Eclipse IDE with Selenium WebDriver. In simple terms, we will create a new Java Project in Eclipse and load all the essential jar files in order to create Selenium Test Scripts.

- Launch Eclipse IDE
- Create a new Java Project from **File > New > Java Project**.

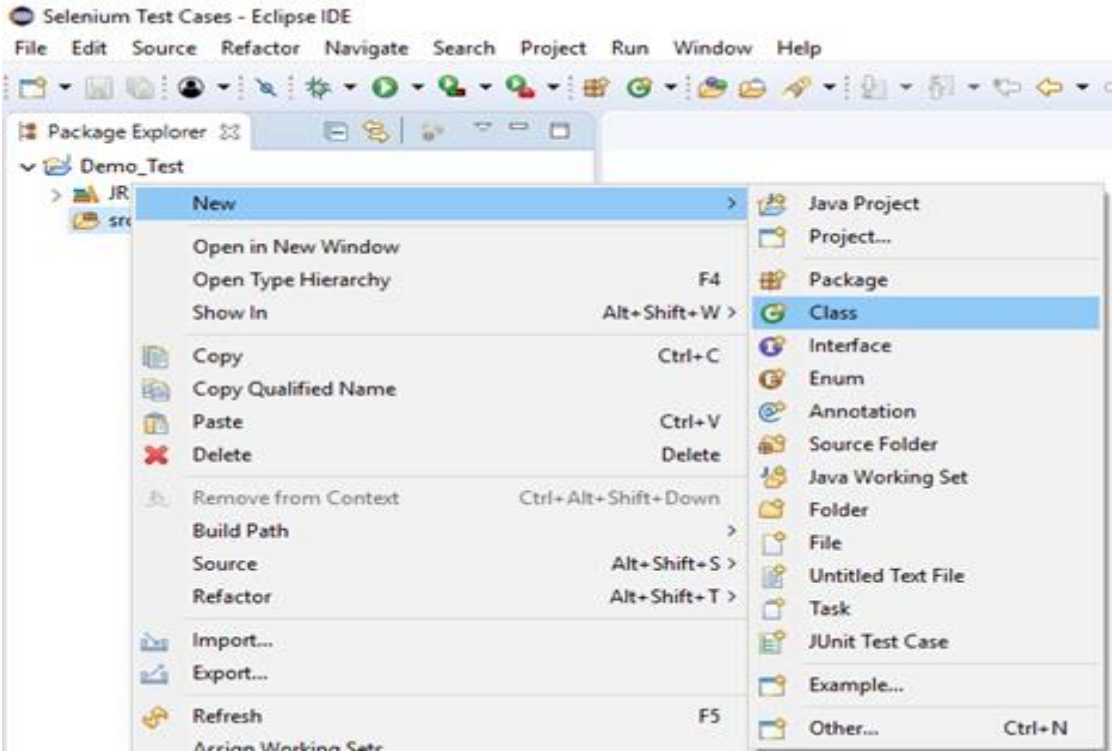


- Give your Project name as "Demo_Test", leave the other fields unaltered and click on "Finish" button.
- It will create a new Java project with the following directories.

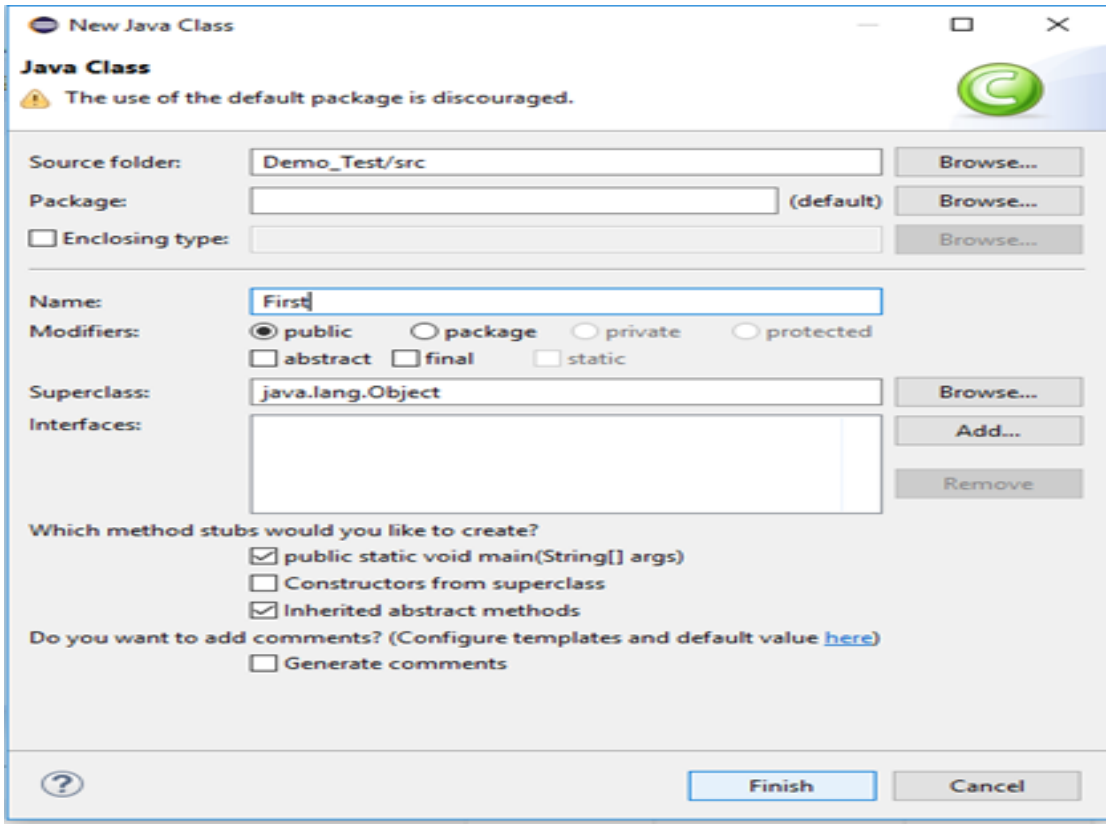


Note: Selenium Test Scripts are always written in ".class" file in Java. Here the project "Demo_Test" act as a Test Suite that may contain one or more Selenium test cases/test scripts.

- Right click on the "src" folder and create a new Class File from **New > Class**.

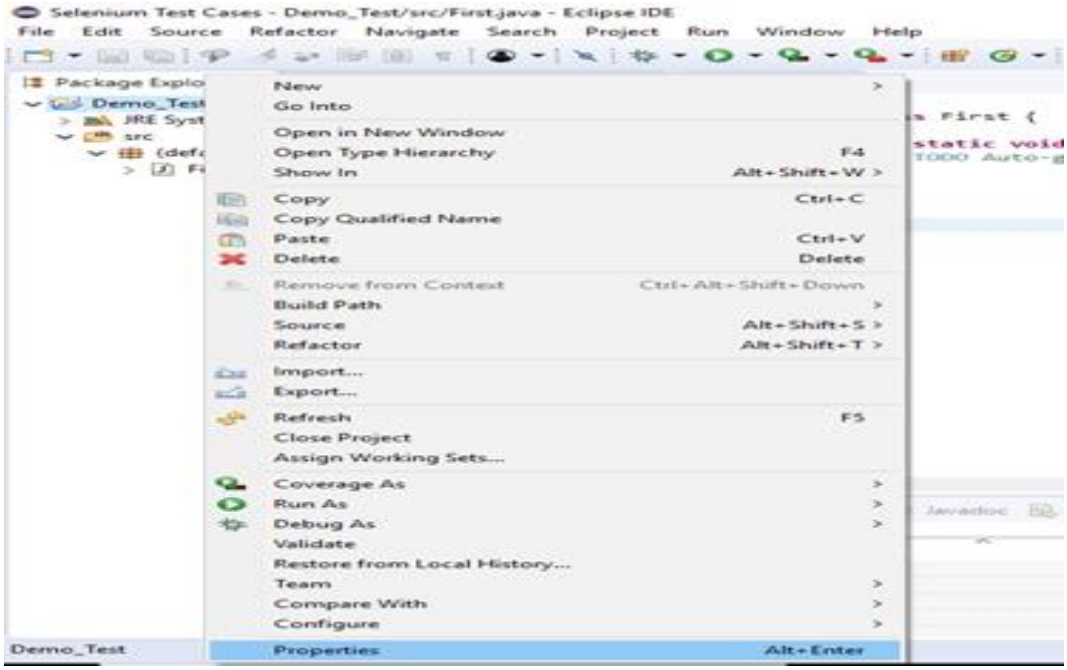


- Give your Class name as "First" and click on "Finish" button.

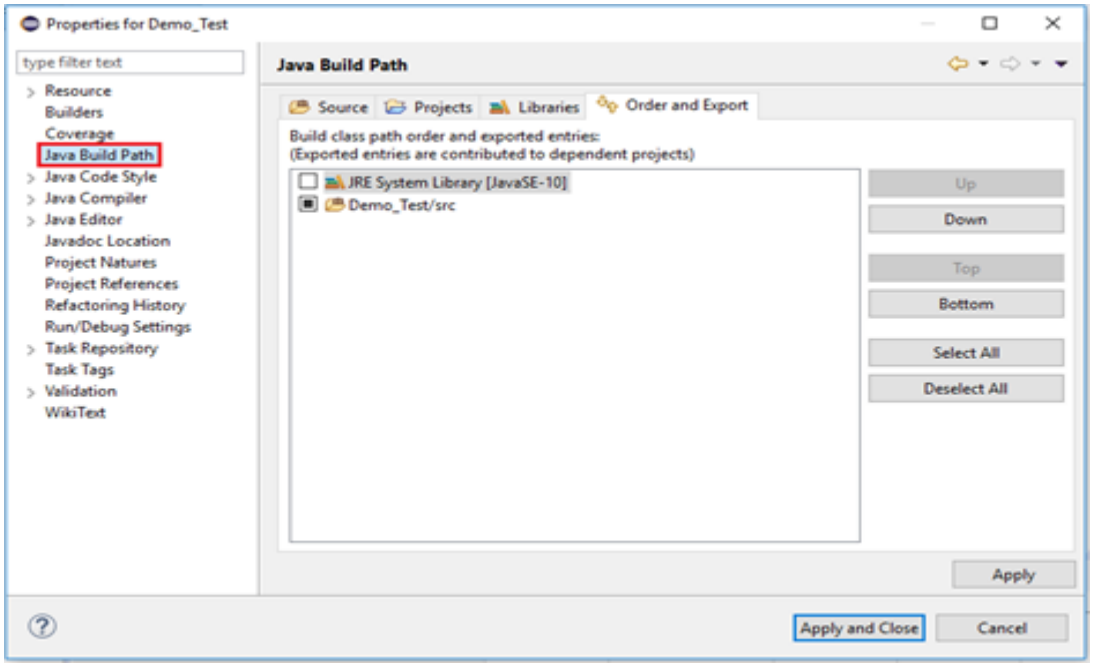


Now, we will add the Selenium jar files in our Test Suite (Demo_Test).

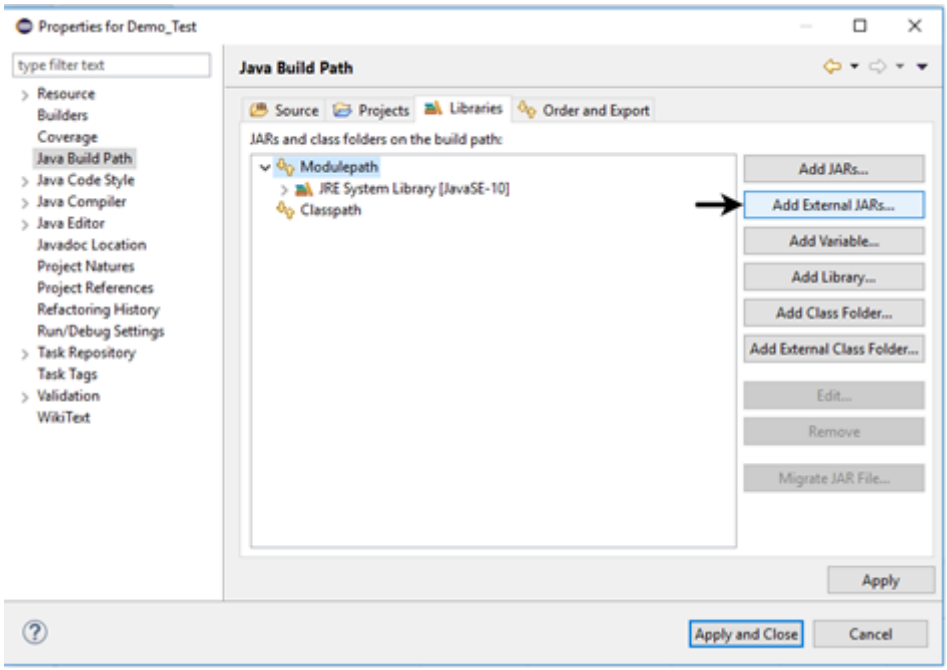
- Right click on "Demo_Test" folder and select Properties.



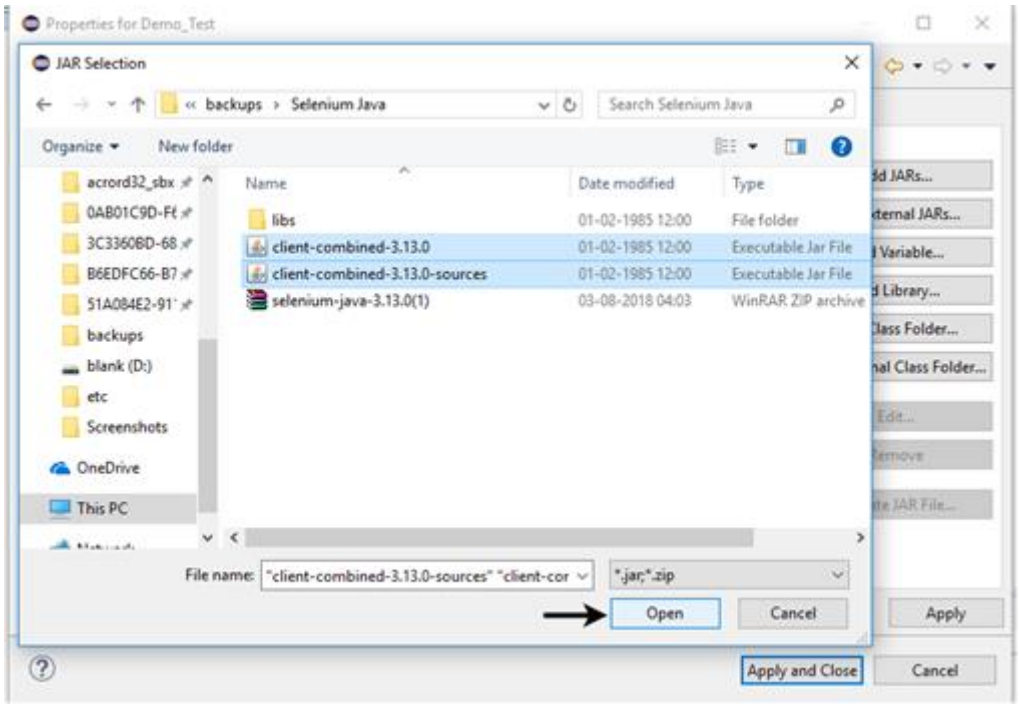
- It will launch the Properties window for our "Demo_Test" Test Suite.
- Click on "Java Build Path" option from the left hand side panel.



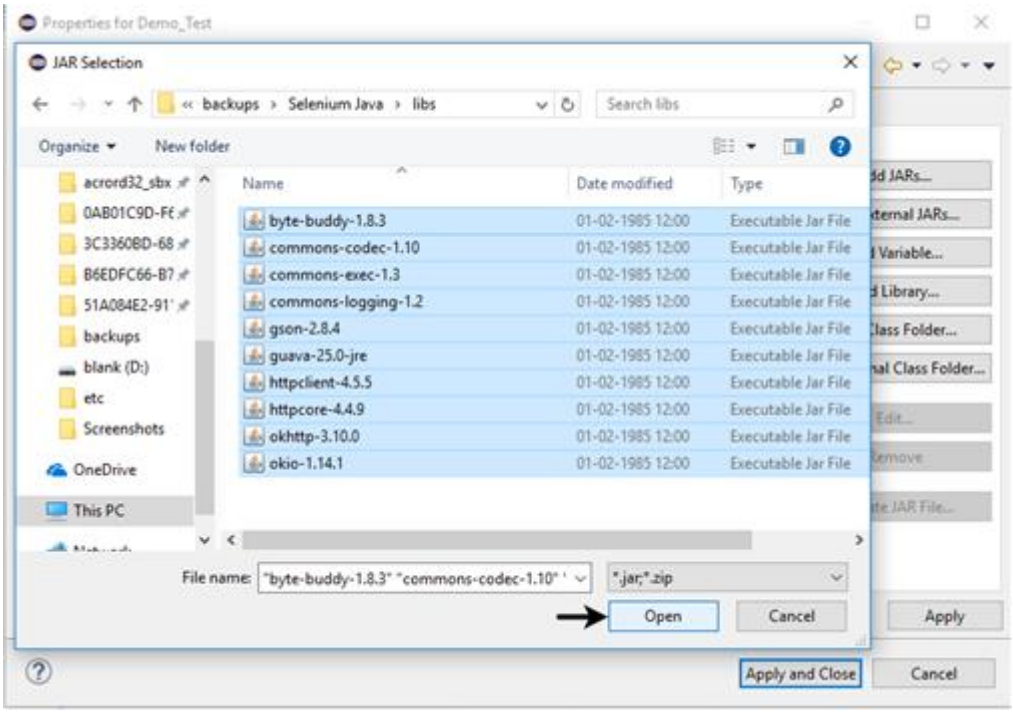
- Switch to Libraries tab and click on "Add External JARs" button.



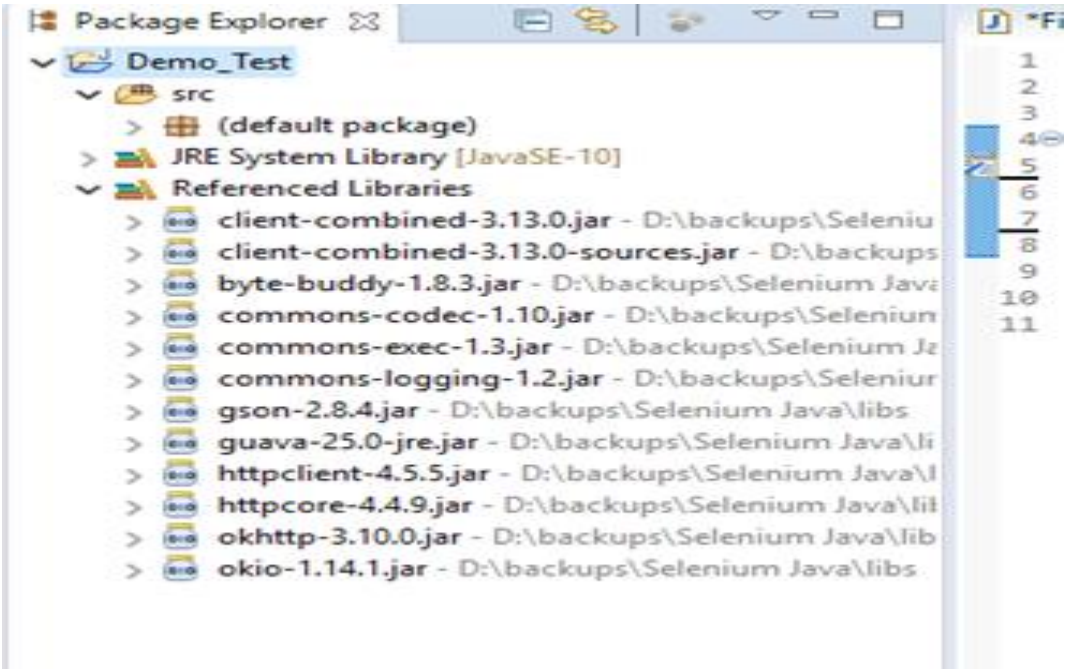
- Locate the directory where you have downloaded the Selenium jar files, select the respective jars and click on "Open" button.



- Repeat the same steps for the jars which are present under the "libs" folder.
- Open "libs" folder, select all of the respective jar files and click on "Open" button.



- Once you get all the Selenium jar files in your Libraries tab, click on Apply and Close button.
- The following image shows the directory structure of our "Demo_Test" test suite after adding Selenium jars.



Conclusion: -

Hence, we have successfully configured Selenium WebDriver with Eclipse IDE. Now, we are ready to write our test scripts in Eclipse and run it in WebDriver.