

Lab Assignment No. 01

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
2	2	2	2	2	10	

Date of Performance: _____

Date of Completion _____

Software Testing And Quality Assurance (STQA) , Group A , Lab Assignment 1

Problem Statement:

Write TEST Scenario for Gmail Login Page

Objectives:

1. To learn about the basic concepts of test scenario.
2. Samples statements of test scenario.

Theory:

Test Scenario is the function that can be tested. It is also called **Test Condition** or **Test Possibility**.

A single Test Scenario can cover one or more Test Cases.

Test Scenario is ‘**What to be tested**’ and **Test Case** is ‘**How to be tested**’.

As a tester, you can put yourself as an end-user and find real situations and possible functionalities of the application being tested.

Why Write Test Scenarios?

Test Scenarios are created for the following reasons:

- To ensure the completion of test coverage.
- Validate that the software is working properly for each Use Case.
- Improve the User Experience.
- Identify the most important end-to-end transactions.
- Study the terminal function and help to build test cases.

Who and When Write Test Scenarios?

Typically, Testers are the ones who are responsible for creating Test Scenarios. In case of some complex and critical applications like Banking Software's, Business Analysts or Test Leads provide the Test Scenarios to Testers. But again depends, and vary from organization to organization.

As, Test Scenarios tell us what needs to be tested, therefore always written before test cases.

Steps to writing Test Scenarios:

As a tester, you can follow these five steps to create a test scenario:

1. Go through all the requirement documents available like BRD, SRS, and FSD to understand the functionalities of the application to be tested.
2. For each requirement, find out possible actions and goals of the user.
3. List down all the possible functionalities/scenarios that need to consider for each requirement. (Test Scenarios Template is provided to download)
4. Once all possible test scenarios are listed, create a Traceability Matrix to ensure that all requirements have a corresponding test scenario.
5. Review the Test Scenario document and Traceability Matrix with Test Lead / Business Analyst.

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

Functional Test Cases :

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 text.	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- Case	Negative/ Positive	Test
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	

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.

Conclusion:

Test Scenario for Gmail Login page written successfully.

Review Questions

- 1. What do you understand by software testing?**
- 2. What is a test plan and what does it include?**
- 3. What are unit testing and integration testing?**
- 4. Mention the different types of software testing.**
- 5. If a product is in the production stage and one of its modules gets updated, then is it necessary to retest it?**

Lab Assignment No. 02

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
2	2	2	2	2	10	

Date of Performance: _____

Date of Completion _____

**Software Testing And Quality Assurance (STQA) ,
Group A , Lab Assignment 2**

Problem Statement:

TEST Scenario for Gmail Login Page

Objectives:

1. To write the test cases for Gmail login page.
2. To TEST written scenario.

Theory :

A test case is a defined format for software testing required to check if a particular application/software/module is working or not. Here we check for different conditions regarding the same.

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.

Why Write Test Cases?

Test cases are one of the most important aspects of software engineering, as they define the way in which the testing would be carried out. Test cases are carried out for a very simple reason, to check if the software actually works or not. There are many advantages of writing test cases:

- Test cases help to check if a particular module/software is meeting the specified requirement or not.
- Test cases determine if a particular module/software work with a given set of conditions.
- Test cases help to narrow down the software needs and required updates.
- Test cases are easy, simple, and clear as they are step by step and well documented.
- Test cases are detailed which makes them helpful during the maintenance phase.

Criteria's To TEST cases :

1. **Simple and clear:** Test cases need to be very concise, clear, and transparent. They should be easy and simple to understand not only for oneself but for others as well.
2. **Uniqueness:** While writing the test cases, it's necessary to make sure that they aren't being written over and over again and each case is different from the other.
3. **No Assumptions:** Test cases should not contain assumed data, don't come up with features/modules that don't exist.

- 4. Traceability:** Test cases should be traceable for the future reference, so while writing it's important to keep that in mind.
- 5. Different input data:** While writing test cases, all types of data must be taken into consideration.
- 6. Strong module name:** The module name should be self-explanatory while writing the test case.
- 7. Minimal Description:** The description of a test case should be small, one or two lines are normally considered good practice but it should give the basic overview properly.
- 8. Maximum conditions:** All kinds of conditions should be taken into consideration while writing a test, increasing the effectiveness.
- 9. Meeting requirements:** While writing the test case it's important that the client/customer/end-user requirements are met.
- 10. Repetitive Results:** The test case must be written in such a way that it should provide the same result.
- 11. Different Techniques:** Sometimes testing all conditions might not be possible but using different testing with different test cases could help to check every aspect of software.

Testing Scenario For Gmail Login Page :

Test ID	Test Description	Test Steps	Test Input	Expected Result	Actual Result	Result	Remark
1.	Verify if a user will be able to login with a valid username and valid password.	1)Enter Valid Username. 2)Enter Valid Password	1)Username: Jivan7855 2)Password: Jivan@1234	User should be logged in.	User logged in successfully.	Pass	Ok
2.	Verify if a user cannot login with a valid username and an invalid password.	1)Enter Valid Username. 2)Enter Invalid Password	1)Username: Jivan7855 2)Password: Jiv8907	User should not be logged in.	User hasn't logged in .	Pass	Ok
3.	Verify the login page for both, when the field is blank and Submit button is clicked.	1)Don't enter username or password	1)Username: 2>Password:	User should get dialogue with message showing that enter username & password	User hasn't logged in and seen the dialogue with enter username and password	Pass	Ok

Conclusion :

Tested Scenario's for Gmail Login Page successfully.

Review Questions

1. What is mean by Test scenarios? Explain with example.
2. Write down comparison between Test scenario & Test Cases.
3. What are the positive & negative test cases for G-mail login page?
4. Test Scenarios for the Sign-up page
5. Test Scenarios for the Login page of Mobile Application

Lab Assignment No. 03

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
2	2	2	2	2	10	

Date of Performance: _____

Date of Completion _____

**Software Testing And Quality Assurance (STQA) ,
Group A , Lab Assignment 3**

Problem Statement:

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

Objectives:

1. To write the test cases for social media application or website.
2. To create an excel sheet for test cases.

Theory :

Standard Fields of a Sample Test Case Template:

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

Your Company LOGO	Project Name:		Test Designed by:	
	Module Name:		Test Designed date:	
	Release Version:		Test Executed by:	
			Test Execution date:	
Pre-condition				
Dependencies:				
Test Priority				
Test Case#	Test Title	Test Summary	Test Steps	Test Data

- 1) **Test case ID:** Unique ID is required for each test case. Follow some conventions to indicate the types of the test.
- 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.
- 10) **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.
- 11) **Dependencies:** Mention any dependencies on other test cases or test requirements.
- 12) **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.
- 13) **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.

14) 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.

15) Post-condition: What should be the state of the system after executing this test case.

16) Actual result: The actual test result should be filled after test execution. Describe the system behavior after test execution.

17) 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**.

18) 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.

Excel sheet for test cases of social media site META :

Test-case-template.xls (1) [Compatibility Mode] - Microsoft Excel					
	A	B	C	D	E
1	META	Project Name:	META	Test Designed by:	XYZ
2		Module Name:	META Login	Test Designed date:	22-09-22
3		Release Version:	META 43.0	Test Executed by:	PQR
4				Test Execution date:	30-09-22
5					
6	Pre-condition	User should have account on META site			
7	Dependencies:				
8	Test Priority	High			
9					
10	Test Case#	Test Title	Test Summary	Test Steps	Test Data
11	TC001	1)verify that user can login or not	1)Valid Username	1)Enter valid username	1)Username:jivan7855
12			2)Valid Password	2)Enter valid Password	2)Password:jivan@123
13					
14			1)Valid Username	1)Enter valid username	1)Username:jivan7855
15			2)Invalid Password	2)Enter invalid Password	2)Password:jivan@125
16					
17			1)Invalid Username	1)Enter invalid username	1)Username:jivan
18			2)Valid Password	2)Enter valid Password	2)Password: jivan@123
19					
20			1)Invalid Username	1)Enter invalid username	1)Username:jivan
21			2)Invalid Password	2)Enter invalid Password	2)Password:jivan098
22					
23					
24					
25					
26					
27					

Test-case-template.xls (1) [Compatibility Mode] - Microsoft Excel

Test Case Template Example						
	Test Data	Expected Result	Post-condition	Actual Result	Status	Remark
11	1)Username:jivan7855					
12	2)Password:jivan@123	User should logged in to his/her account	User should login successfully	User logged in successfully	Pass	Ok
13						
14						
15	1)Username:jivan7855					
16	2)Password:jivan@125	User should see error message as enter valid username and password	User should not be logged in	message has shown to user that is as enter valid username and password	Pass	Ok
17						
18	1)Username:jivan					
19	2)Password:jivan@123	User should see error message as enter valid username and password	User should not be logged in	message has shown to user that is as enter valid username and password	Pass	Ok
20						
21	1)Username:jivan					
22	2)Password:jivan098	User should see error message as enter valid username and password	User should not be logged in	message has shown to user that is as enter valid username and password	Pass	Ok
23						
24						
25						
26						
27						
28						
29						
30						
31						

Conclusion:

Test cases for social media site written successfully.

Review Questions

1. What is Test Case? Who prepares Test Cases?
2. What Are Basic Types Of Test Cases?
3. How to write good Test cases?
4. Write A Test Case For Telephone?
5. Write Test Cases For This Scenario If A Job Fails It Should Get Restarted Again This Should Happen For Three Times If It Fails Again Then It Should Quit?

Lab Assignment No. 04

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
2	2	2	2	2	10	

Date of Performance: _____

Date of Completion _____

**Software Testing And Quality Assurance (STQA) ,
Group A , Lab Assignment 4**

Problem Statement:

Create Defect Report for Any application or web application

Objectives:

1. To learn about defect report.
2. To create defect report for an 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.

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).
- 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.

Bug report sample : Web Application :-

Summary: In CTR (Click through ratio) ‘Total’ row calculation is wrong

Product Name : Photo Editor

Version: 1.0

Device : PC

Platform : Windows OS

URL: <https://photoedits.in/xnp.html>

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

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.

Conclusion :

Defect report for an application created successfully.

Review Questions

1. Differentiate Error, Defect, and Failure?
2. Give a sample Defect report template?
3. What is Defect Management? What is Defect Management Tool?
4. What are the different types of Status of Defects?
5. What do you do if the defect is rejected?

Lab Assignment No. 05

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
2	2	2	2	2	10	

Date of Performance: _____

Date of Completion _____

Software Testing And Quality Assurance (STQA) , Group A , Lab Assignment 5

Problem Statement:

Installation Of Selenium Grid And Selenium Webdriver & Java Eclipse
(Automation Tools).

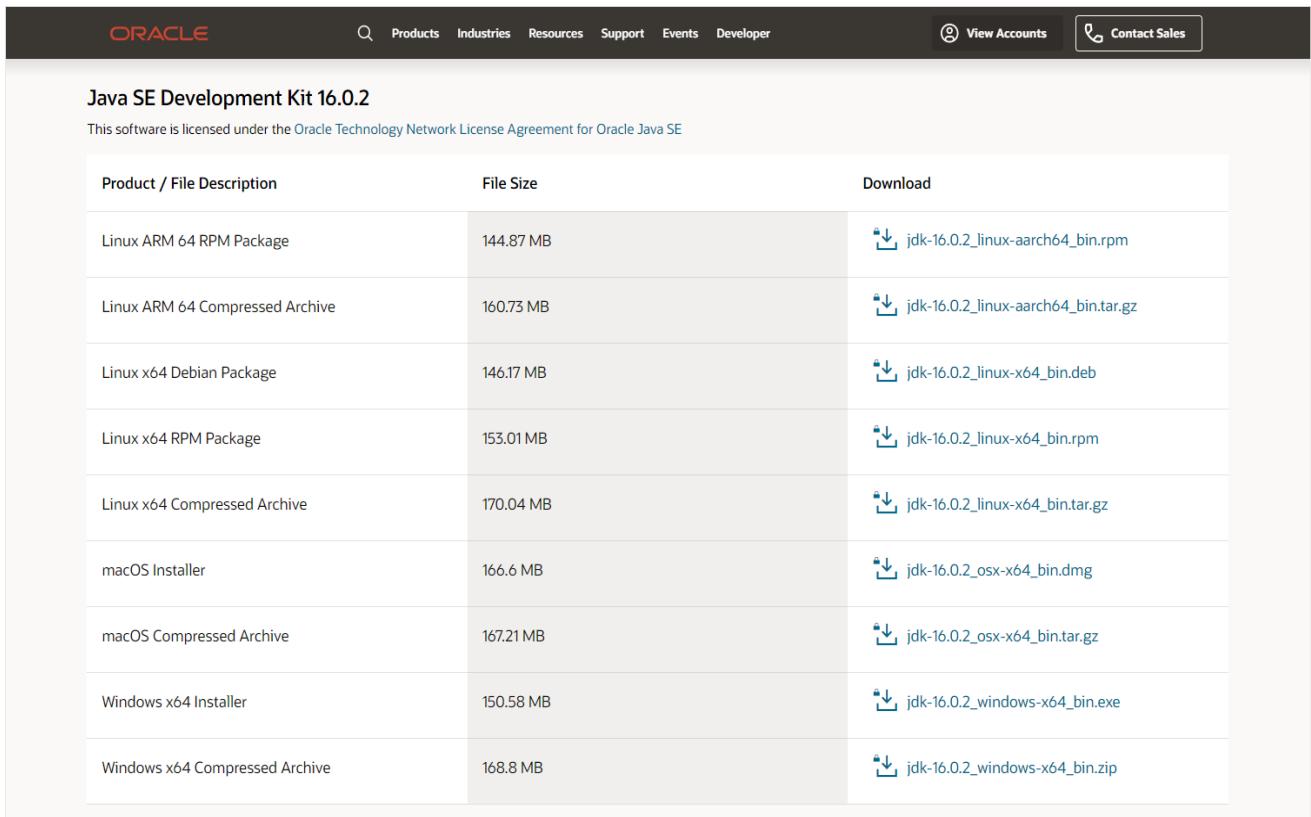
Objectives:

1. To install automation tools.

Theory :

Prerequisites for configuring Selenium in Eclipse :

- **Install Java** Download Java SE Development Kit 16.0.2 according to the Windows, Linux, or macOS platform used.



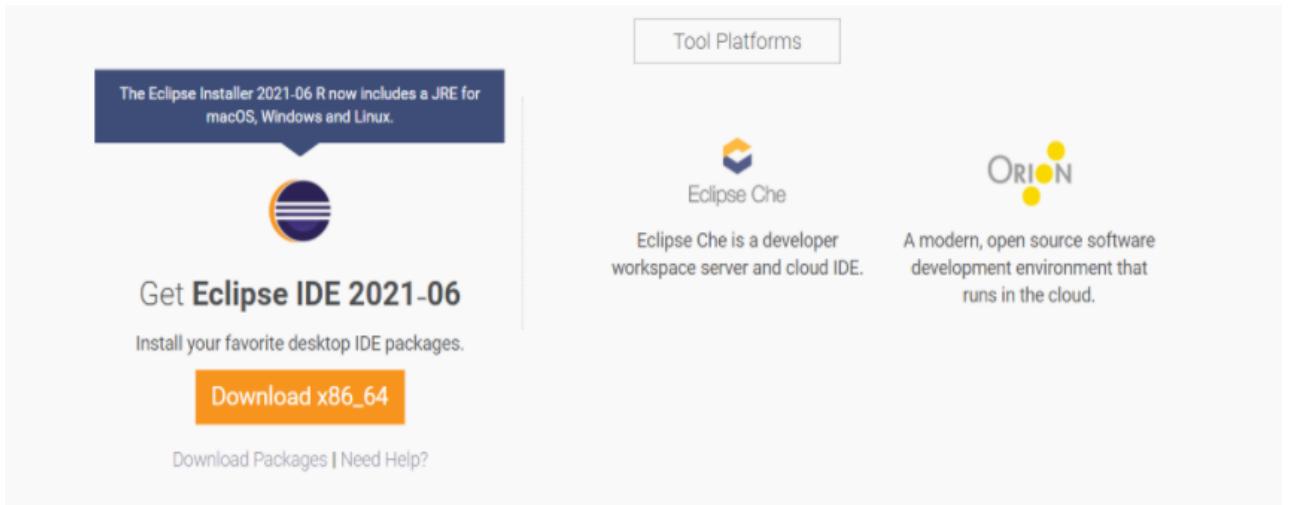
The screenshot shows the Oracle Java SE Development Kit 16.0.2 download page. At the top, there's a navigation bar with links for Products, Industries, Resources, Support, Events, Developer, View Accounts, and Contact Sales. Below the navigation, the title "Java SE Development Kit 16.0.2" is displayed, followed by a note: "This software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE". A table lists nine download options:

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	167.21 MB	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

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

jdk.exe /s.

- **Install Eclipse IDE**



The screenshot shows the Eclipse IDE 2021-06 download page. It features a banner stating "The Eclipse Installer 2021-06 R now includes a JRE for macOS, Windows and Linux." Below the banner is the Eclipse logo. To the right, there's a "Tool Platforms" section featuring the Eclipse Che logo and text about it being a developer workspace server and cloud IDE. Further to the right is the Orion logo with a description of it being a modern, open source software development environment that runs in the cloud. At the bottom left, there's a call-to-action button "Download x86_64" and links for "Download Packages" and "Need Help?".

- **Install Selenium Grid**

Download and Install Selenium to be set up in Eclipse.

- **Install Selenium Driver**

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

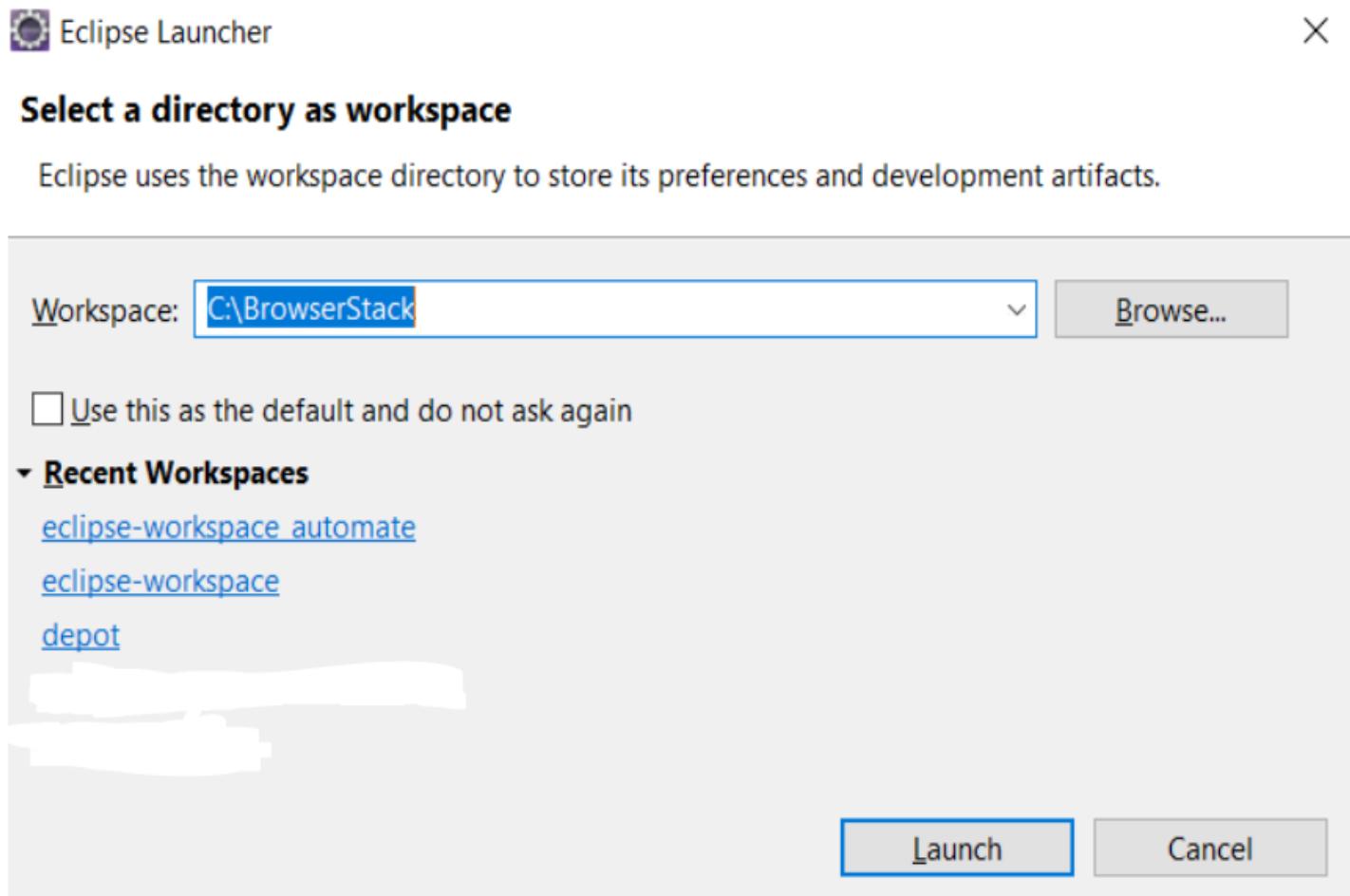
How to configure Selenium in Eclipse :

Step 1: Launch Eclipse

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

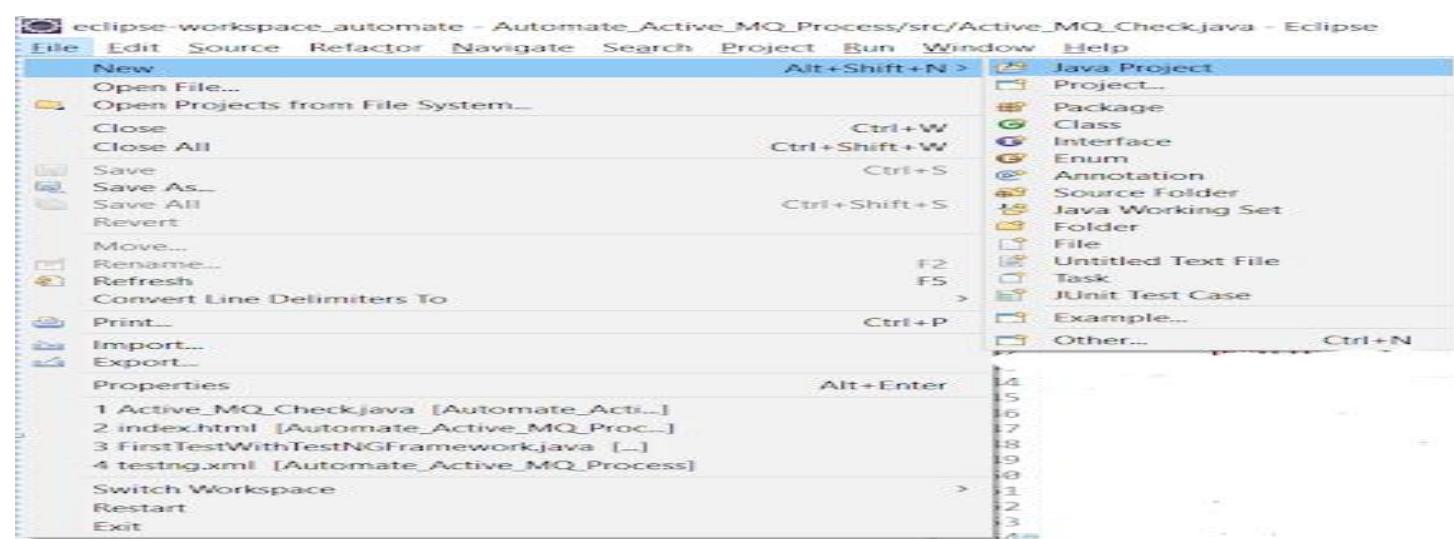
Step 2: Create Workspace in Eclipse

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



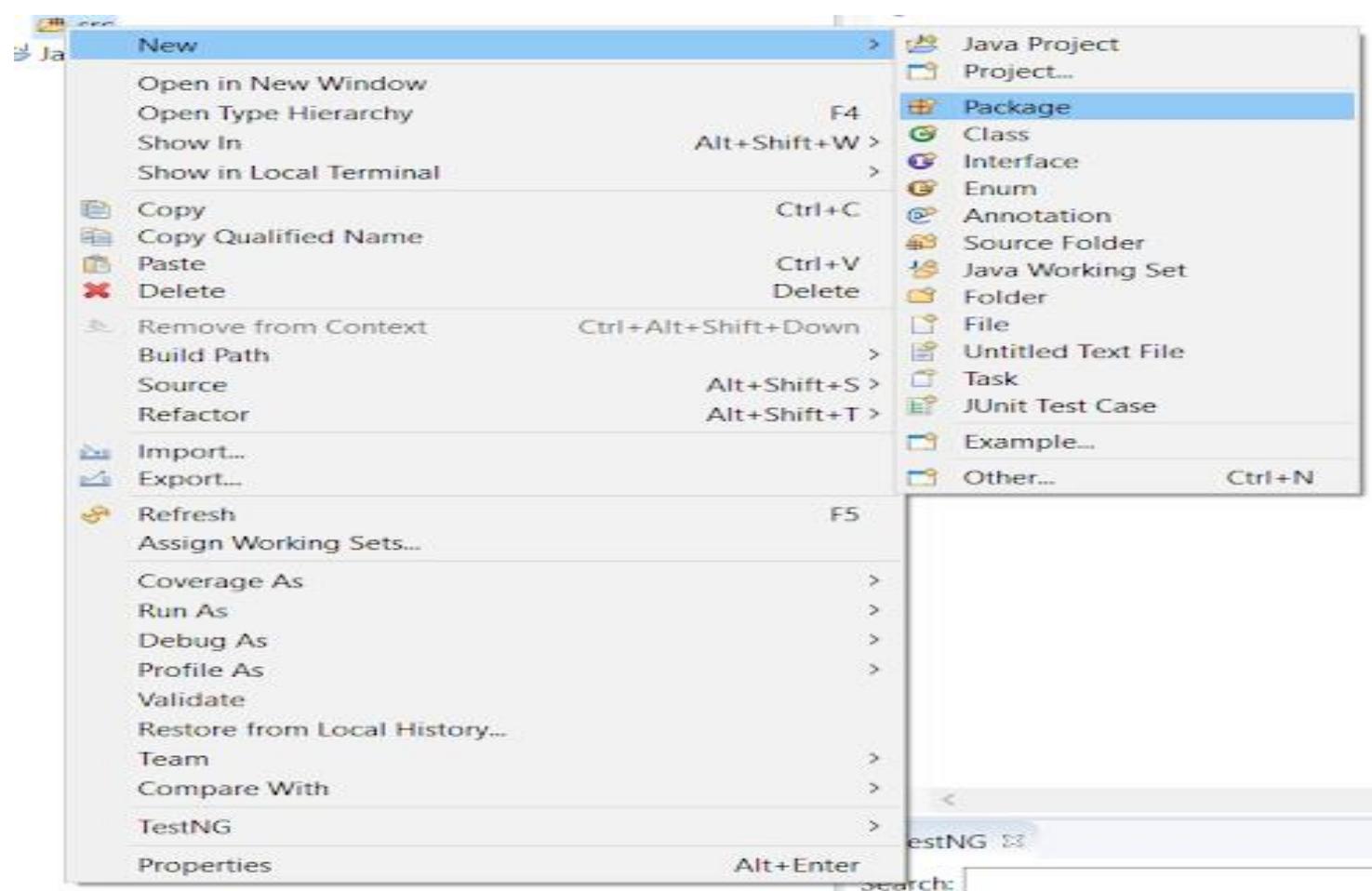
Step 3: Create New Java Project in the Browser Stack Workspace

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



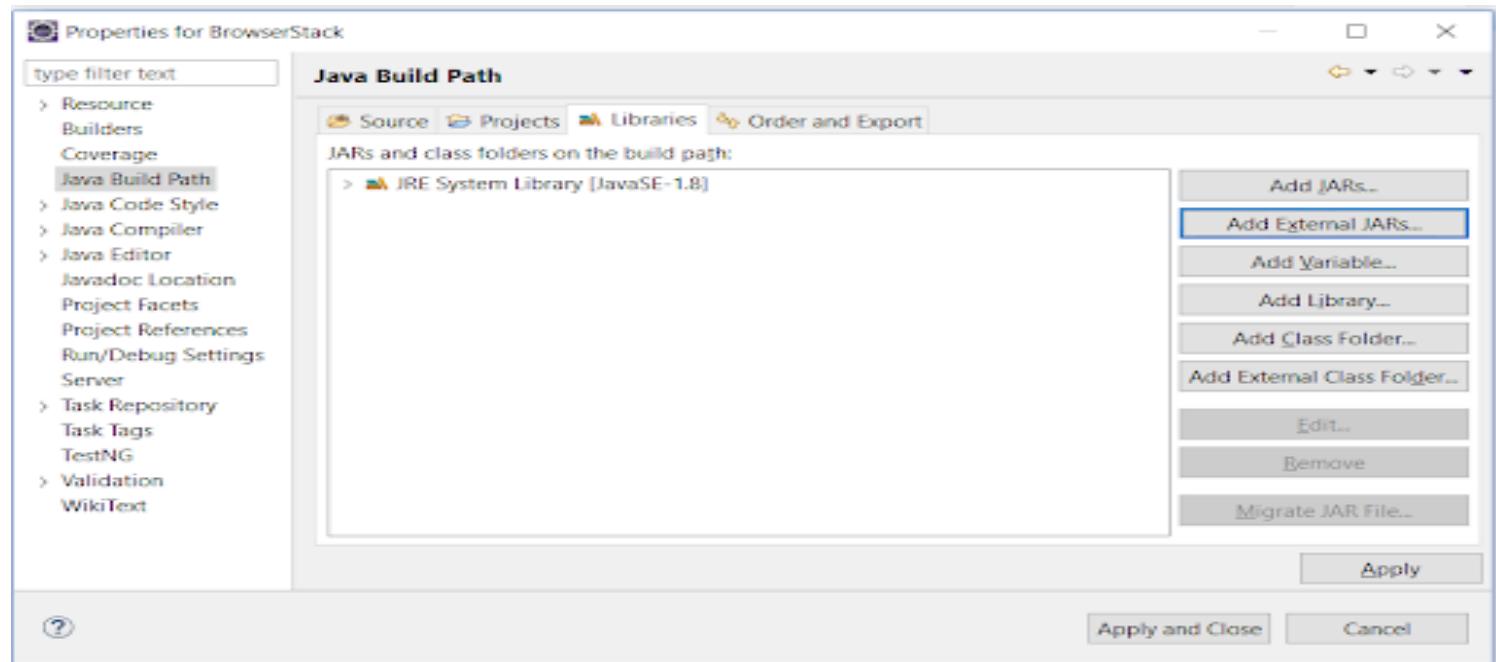
Step 4: Create Package and Class under the Java Project

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.



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**. 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

Installation of automation tools is done successfully

Review Questions

1. What are the limitations of Selenium testing?
2. What makes Selenium such a widely used testing tool? Give reasons.
3. How to select a value from a dropdown in Selenium WebDriver?
4. What is test automation or automation testing? What are the advantages of automation testing?
5. Name some of the commonly used automation testing tools that are used for non-functional automation.

Lab Assignment No. 06

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
2	2	2	2	2	10	

Date of Performance: _____

Date of Completion _____

Software Testing And Quality Assurance (STQA) , Group A , Lab Assignment 6

Problem Statement:

Prepare Software requirement specification for any project or problem statement

Objectives:

1. To learn about software requirement specification.
2. To prepare SRS for project.

Theory :

Software Requirement Specification (SRS) Format : as name suggests, is complete specification and description of requirements of software that needs to be fulfilled for successful development of software system. These requirements can be functional as well as non-functional depending upon type of requirement. The interaction between different customers and contractor is done because its necessary to fully understand needs of customers.

Depending upon information gathered after interaction, SRS is developed which describes requirements of software that may include changes and modifications that is needed to be done to increase quality of product and to satisfy customer's demand.

1) Introduction :

- a) **Purpose of this Document** – At first, main aim of why this document is necessary and what's purpose of document is explained and described.
- b) **Scope of this document** – In this, overall working and main objective of document and what value it will provide to customer is described and explained. It also includes a description of development cost and time required.
- c) **Overview** – In this, description of product is explained. It's simply summary or overall review of product.

2) **General description** : In this, general functions of product which includes objective of user, a user characteristic, features, benefits, about why its importance is mentioned. It also describes features of user community.

3) **Functional Requirements** : In this, possible outcome of software system which includes effects due to operation of program is fully explained. All functional requirements which may include calculations, data processing, etc. are placed in a ranked order.

4) **Interface Requirements** : In this, software interfaces which mean how software program communicates with each other or users either in form of any language, code, or message are fully described and explained. Examples can be shared memory, data streams, etc.

5) **Performance Requirements** : In this, how a software system performs desired functions under specific condition is explained. It also explains required time, required memory, maximum error rate, etc.

6) **Design Constraints** : In this, constraints which simply means limitation or restriction are specified and explained for design team. Examples may include use of a particular algorithm, hardware and software limitations, etc.

- 7) Non-Functional Attributes :** In this, non-functional attributes are explained that are required by software system for better performance. An example may include Security, Portability, Reliability, Reusability, Application compatibility, Data integrity, Scalability capacity, etc.
- 8) Preliminary Schedule and Budget :** In this, initial version and budget of project plan are explained which include overall time duration required and overall cost required for development of project.
- 9) Appendices :** In this, additional information like references from where information is gathered, definitions of some specific terms, acronyms, abbreviations, etc. are given and explained.

Software Requirements Specification For eCommerce:

1) Introduction :

The purpose of this document is to outline the requirements for the eCommerce (Business to Customer) Product to be developed for IBEE Solutions (P) Ltd.

2) Overall description :

- 2.1 Product Perspective
- 2.2 Product Functions
- 2.3 User Classes and Characteristics
- 2.4 Operating Environment
- 2.5 Design and Implementation Constraints
- 2.6 User Documentation
- 2.7 Assumptions and Dependencies

3) External Interface Requirements :

- 3.1 User Interfaces
- 3.2 Hardware Interfaces
- 3.3 Software Interfaces
- 3.4 Communication Interfaces

4) Functional Requirement Specifications (FRS) :

- 4.1 System Features
- 4.2 Functional Requirements

5) Non Functional Requirements

- 5.1 Usability Requirements

5.2 Performance Requirements

5.3 Compatibility Requirements

6) Other Requirements

7) Glossary

Conclusion:

Prepared the software requirement specification document for an eCommerce business.

Review Questions

1. What Is SRS In Project?
2. What Is The Need For An SRS Document?
3. How Many Types Of Software Requirements Are There? Specify Them?
4. What Are The Software Requirement Validations?
5. Explain Functional And Non Functional Requirements?