

Laboratory Practice-IV

410245(D): Software Testing and Quality Assurance

Any 5 assignments from group 1 and 1 Mini project from group 2 is mandatory

Group 1:

1. Write TEST Scenario for Gmail Login Page
2. Writ Test Scenario for Gmail Login Page
3. Write Test cases in excel sheet for Social Media application or website 4. Create Defect Report for Any application or web application
5. Installation of Selenium grid and selenium Web driver java eclipse (automation tools).
6. Pre – Prepare Software requirement specification for any project or problem statement

Group 2:

7. Mini Project: Software Testing and Quality Assurance Mini Project Dynamic website of covid19 information using HTML, CSS, JAVASCRIPT And PHP, MySQL database used to store user account, comment, and registration form details. Regular Expression testcases for testing purpose
8. Mini Project: 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
9. Mini Project: 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

Assignment No : 1

Write TEST Scenario for Gmail Login Page

Test Cases of a Login Page (Test Scenarios Login Page):

Following are the test cases for User Login Page. The list consists of both Positive and Negative test scenarios of login page along with UI test cases, Security test cases and so on.

UI Test Scenarios of Login Page

1. Verify that the login screen contains elements such as Username, Password, Sign in button, Remember password check box, Forgot password link, and create an account link.
2. Verify that all the fields such as Username, Password has a valid placeholder
3. Verify whether all the text boxes have a minimum and maximum length.
4. Verify that the labels float upward when the text field is in focus or filled (In case of the floating label).
5. Verify to see if the font style and size of the labels, as well as the text on each object, are clearly visible.
6. Verify that the application's user interface (UI) is responsive, so it will adapt to different screen resolutions and devices.
7. Verify the login page and all the fields in the login page are displaying without any break in different browsers.

Functional Test Scenarios of Login Page

1. Verify that cursor is focused on the "Username" text box on the page load (login page)
2. Verify that tab functionality is working properly or not
3. Verify that Enter/Tab key works as a substitute for the Sign-in button
4. Verify that the User is able to Login with Valid Credentials
5. Verify that the User is not able to Login with an invalid Username and invalid Password
6. Verify that the User is not able to Login with a Valid Username and invalid Password

7. Verify that the User is not able to log in with an invalid Username and Valid Password
8. Verify that the User is not able to log in with a blank Username or Password
9. Verify that the User is not able to Login with inactive credentials
10. Verify that the reset button clears the data from all the text boxes in the login form
11. Verify that the login credentials, mainly password stores in a database in an encrypted format
12. Verify that clicking on the browser back button after successful login should not take the User to log out mode
13. Verify that validation message is displayed in the case when User leaves Username or Password as blank
14. Verify that validation message is displayed in case of exceeding the character limit of the Username and Password fields
15. Verify that validation message is displayed in case of entering special character in the Username and password fields
16. Verify that the “Keep me logged in” checkbox is unselected by default (depends on business logic, it may be selected or unselected)
17. Verify that the timeout of the login session (Session Timeout)
18. Verify that the logout link is redirected to login/home page
19. Verify that User is redirected to appropriate page after successful login
20. Verify that the User is redirected to the Forgot password page when clicking on the Forgot Password link.
21. Verify that the User is redirected to the Create an account page when clicking on the Signup / Create an account link
22. Verify that the User should be able to login with the new password after changing the password
23. Verify that the user should not be able to login with the old password after changing the password
24. Verify that spaces should not be allowed before any password characters attempted

25. Verify whether the user is still logged in after a series of actions such as sign-in, close the browser, and reopen the application.
26. Verify that the ways to retrieve the password if the user forgets the password

Non-functional Security Test Cases for Login Page

1. Verify that clicking on the browser back button after successful logout should not take the User to a logged-in mode
2. Verify that there is a limit on the total number of unsuccessful login attempts (No. of invalid attempts should be based on business logic. Based on the business logic, User will be asked to enter the captcha and try again or user will be blocked)
3. Verify that the password is in encrypted form (masked format) when entered in the password field.
4. Verify the password can be copy-pasted. System shouldn't allow users to copy paste password.
5. Verify that encrypted characters in the "Password" field should not allow deciphering if copied
6. Verify that the "Remember password" checkbox is unselected by default (depends on business logic, it may be selected or unselected)
7. Verify whether the login form is revealing any security information by viewing the page source
8. Verify that the login page is vulnerable to SQL injection.
9. Verify whether Cross-site scripting (XSS) vulnerability works on a login page. XSS vulnerability may be used by hackers to bypass access controls.

Performance Test Cases for Login Page

Verify that how much time the application is taking to load the home page after entering the valid user name and password in the login page.

Test Cases for CAPTCHA & Cookies (If there is a captcha on the login page)

1. Verify that whether there is a client-side validation when the User doesn't enter the CAPTCHA

2. Verify that the refresh link of CAPTCHA is generating the new CAPTCHA
3. Verify that the CAPTCHA is case sensitive
4. Verify whether the CAPTCHA has audio support to listen
5. Verify whether virtual keyboard is available and working properly to enter login credentials incase of banking applications.
6. Verify two-way authentication through OTP is working properly incase of banking applications.
7. Verify SSL certificate is implemented or not
8. Verify that the user is able to login when the browser cookies are cleared. When the cookies are cleared, system should not allow user to login automatically.
9. Verify the login functionality when the browser cookies are turned off.

Assignment No : 2
Write TEST Scenario for Inbox Functionality.

Test Case for Gmail – Inbox Functionality

- Verify that a newly received email is displayed as highlighted in the Inbox section.
- Verify that a newly received email has correctly displayed sender email Id or name, mail subject and mail body(trimmed to a single line).
- Verify that on clicking the newly received email, the user is navigated to email content.
- Verify that the email contents are correctly displayed with the desired source formatting.
- Verify that any attachments are attached to the email and are downloadable.
- Verify that the attachments are scanned for viruses before download.
- Verify that all the emails marked as read are not highlighted.
- 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.
- Verify that count of unread emails is displayed alongside 'Inbox' text in the left sidebar of Gmail.
- Verify that unread email count increases by one on receiving a new email.
- Verify that unread email count decreases by one on reading an email (marking an email as read).
- Verify that email recipients in cc are visible to all users.
- Verify that email recipients in bcc are not visible to the user.
- Verify that all received emails get piled up in the 'Inbox' section and get deleted in cyclic fashion based on the size availability.

Verify that email can be received from non-Gmail email Ids like – yahoo, Hotmail etc.

Assignment No : 3

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

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:

- TestRail makes tracking test results easier.
- It seamlessly gets integrated with bug trackers, automated tests, etc.
- Personalized to-do lists, filters, and email notifications will help with boosting productivity.
- Dashboards and activity reports are for easy tracking and following the status of individual tests, milestones, and projects.

#2) **Katalon Platform**



Katalon Platform 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.

=> **Get Started for Free**

#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:

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

Here is how to make the manual test case management process a bit easier with the help of simple testing templates.

Note: I've listed the maximum number of fields related to the test case. However, it is advised to use only those fields used by your team. Also, if you think any fields used by your team are missing from this list, then feel free to add them to your customized template.

Standard Fields of a Sample Test Case Template

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

Test Case Template						
Project Name:						
Test Case ID: <u>Fun_10</u>			Test Designed by: <u><Name></u>			
Test Priority (Low/Medium/High): <u>Med</u>			Test Designed date: <u><Date></u>			
Module Name: <u>Google login screen</u>			Test Executed by: <u><Name></u>			
Test Title: <u>Verify login with valid username and password</u>			Test Execution date: <u><Date></u>			
Description: <u>Test the Google login page</u>						
Pre-conditions: <u>User has valid username and password.</u>						
Dependencies:						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to login page	User= <u>sample@gmail.com</u>	User should be able to login	User is navigated to	Fail	
2	Provide valid username	Password: <u>1234</u>		dashboard with successful		
3	Provide valid password			login		
4	Click on Login button					
Post-conditions: <u>User is validated with database and successfully login to account. The account session details are logged in database.</u>						

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

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'.

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.

Module Name: Mention the name of the main module or the sub-module.

Test Designed By Name of the Tester.

Test Designed Date: Date when it was written.

Test Executed By Name of the Tester who executed this test. To be filled only after test execution.

Test Execution Date: Date when the test was executed.

Test Title/Name: Test case title. **For example,** verify the login page with a valid username and password.

Test Summary/Description: Describe the test objective in brief.

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.

Dependencies: Mention any dependencies on other test cases or test requirements.

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.

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.

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

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**.

Add the following fields if necessary:

Test Type/Keywords: This field can be used to classify tests based on test types. **For Example,** functional, usability, business rules, etc.

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.

With the help of the above fields, I've prepared an example test case template for your reference.

Download Test Case Template with Example (Format #1)

- Test case DOC file template & Test case Excel file 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

Sample Test Cases:**One More Test Case Format (#2)**

Undoubtedly, the test cases will differ depending upon the functionality of the software that it is intended for. However, given below is a template that you can always use to document the test cases without bothering about what your application is doing.

Test Scenario ID				Test Case ID				
Test Case Description				Test Priority				
Pre-Requisite				Post-Requisite				
Test Execution Steps:								
S.No	Action	Inputs	Expected Output	Actual Output	Test Browser	Test Result	Test Comments	

Sample Test Cases

Based on the above template, below is an **example** that showcases the concept in a much understandable way.

Let's assume that 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-1A				
Test Case Description	Login – Positive test case	Test Priority	High				
Pre-Requisite	A valid user account	Post-Requisite	NA				
Test Execution Steps:							
S.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	[Priya 10/17/2017 11:44 AM]: Launch successful
2	Enter correct Email & Password and hit login button	Email id : test@xyz.com Password: *****	Login success	Login success	IE-11	Pass	[Priya 10/17/2017 11:45 AM]: Login successful

Test Scenario ID		Login-1		Test Case ID		Login-1B	
Test Case Description		Login – Negative test case		Test Priority		High	
Pre-Requisite		NA		Post-Requisite		NA	
Test Execution Steps:							
S.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	[Priya 10/17/2017 11:44 AM]: 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	[Priya 10/17/2017 11:45 AM]: 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	[Priya 10/17/2017 11:46 AM]: Invalid login attempt stopped

Test Case Example for Manual Testing

Below given is an **example** of a live project that demonstrates how all the above-listed tips and tricks are implemented.

[Note: Click on any image for an enlarged view]

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
1												
2	Search	Search-1										
3	Search	Search-2										
4	Search	Search-3										
5	Search	Search-4										
6	Search	Search-5										
7	Search	Search-6										
8	Search	Search-7										
9	Search	Search-8										
10	Search	Search-9										
11												
12	Search	Search-10										
13	Search	Search-11										
14												
15	HRA	HRA-1										
16		HRA-2										
17												

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

Test Scenario Group	Test Case Id	Test Case Description	Test Input	Expected Result	Actual Result	Test Browser	Executed Date	Test Results
Health records								
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		
	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	what does your blood pressure range, on average?	what does your blood pressure range, on average?	Chrome,IE, Firefox		
			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		
	PH 3	Diabetes:	a) Yes and on medication	how old were you at time of dx?		Chrome,Firefox		
			b) Yes but not on medication	how old were you at time of dx?		Chrome,Firefox		
			a) Yes and on medication	what type of dm do you have?		Chrome,Firefox		
			b) Yes but not on medication	what type of dm do you have?		Chrome,Firefox		

Conclusion

Personally, I prefer to use a Test Case Management tool. You can start with an open- source tool. It will be a good addition to your efforts to set up the testing process and meanwhile, it will also save a lot of time instead of manually maintaining these documents.

We have also seen test case templates and a few examples using very good, quality documentation. I hope this article would have been helpful to you.

Assignment No : 4

Create Defect Report for Any application or web application

Sample bug report : 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.

Assignment No : 5

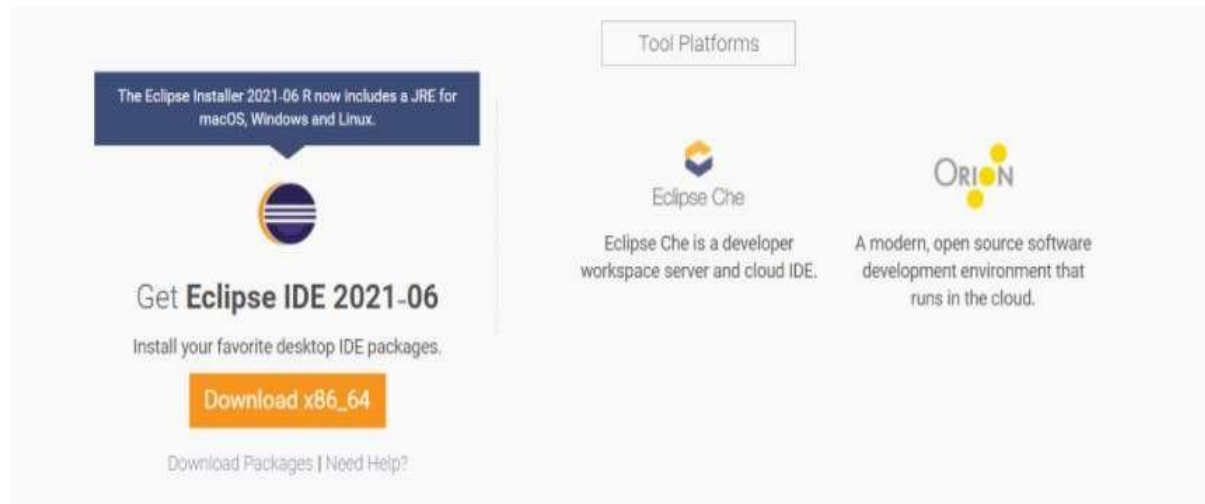
Installation of Selenium grid and selenium Web driver java eclipse (automation tools).

Link : <https://www.browserstack.com/guide/how-to-setup-selenium-in-eclipse>

- **Prerequisites for configuring Selenium in Eclipse**
- **Install Java** Download Java SE Development Kit 16.0.2 according to the Windows, Linux, or macOS platform being used.
[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`

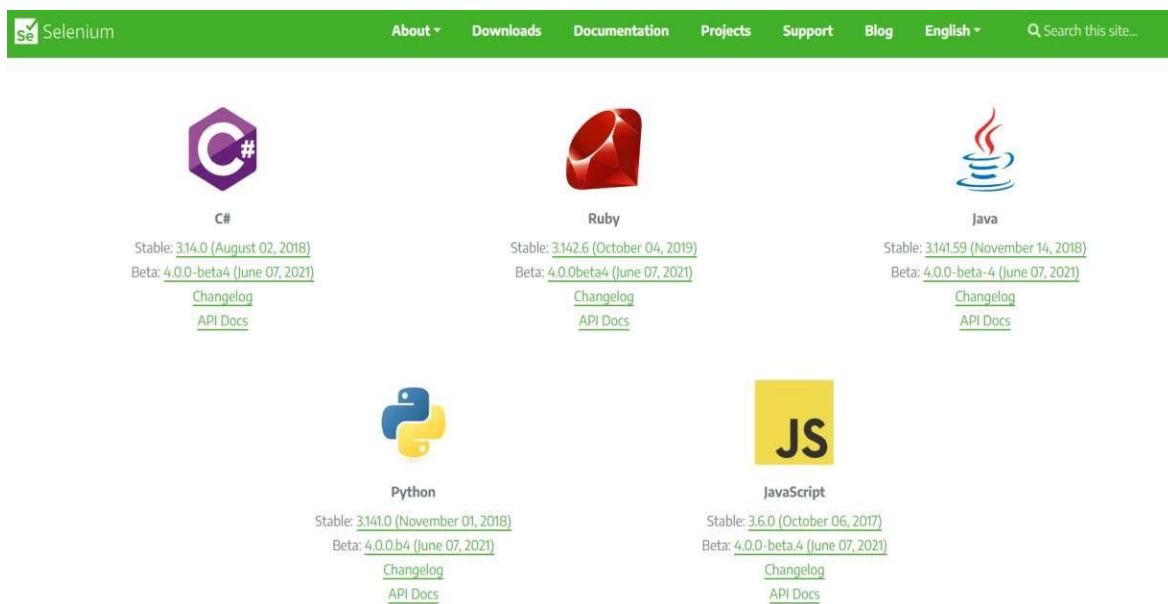
ORACLE			Q	Products	Industries	Resources	Support	Events	Developer	View Accounts	Contact Sales
Java SE Development Kit 16.0.2			This software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE								
Product / File Description	File Size	Download									
Linux ARM 64 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									

Install Eclipse IDE



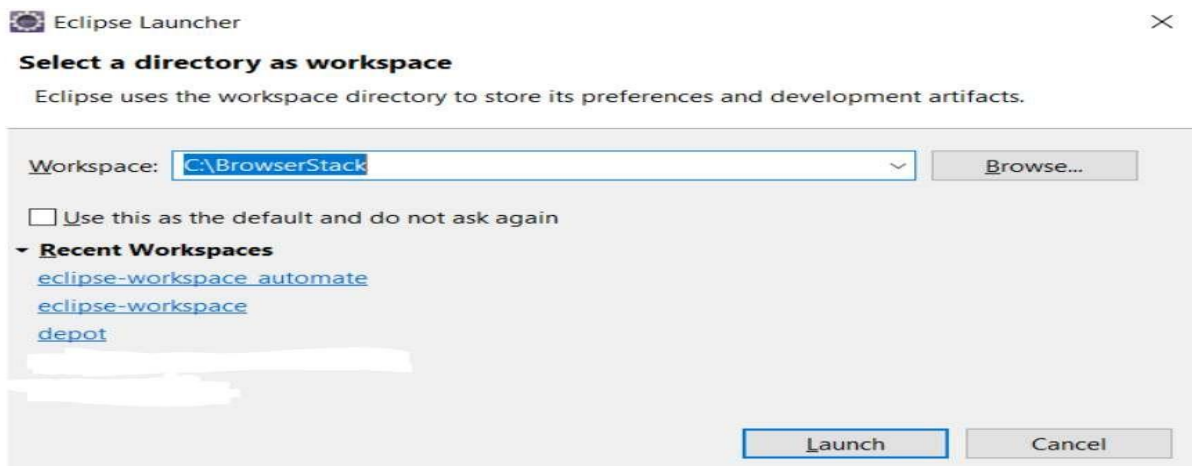
Install Selenium

- 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](#)

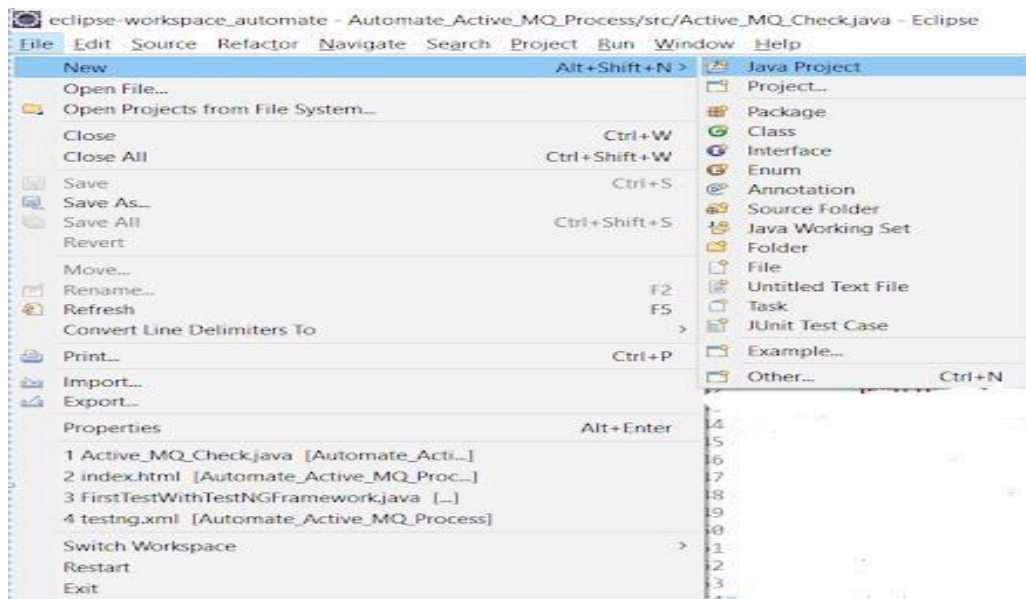


How to configure Selenium in Eclipse

- 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.



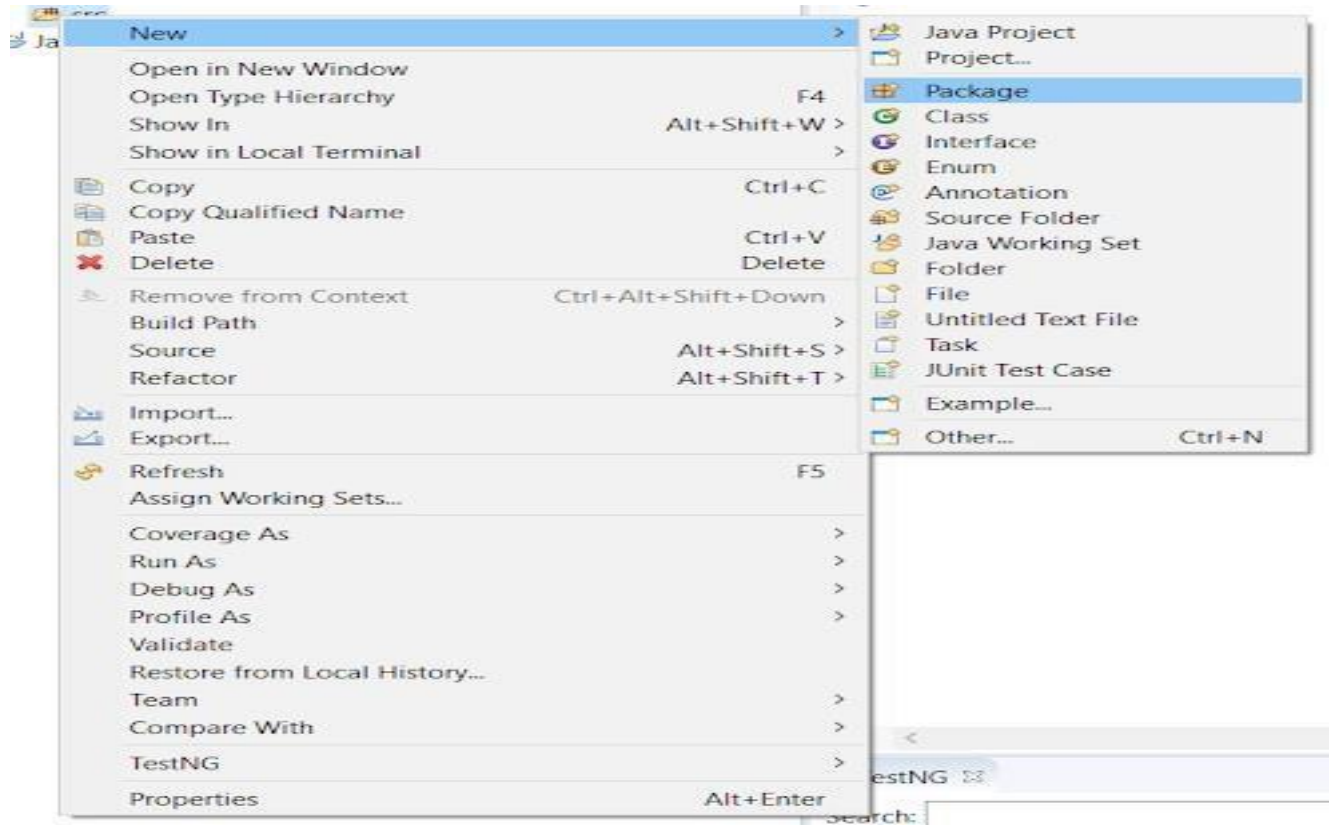
- **Step 3: Create New Java Project in the BrowserStack Workspace**
- Create a new Java Project by clicking on **File > New > Java Project** and name it.



- **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.



Assignment No : 6

Prepare Software requirement specification for any project or problem statement

Brief Requirements Specification: Online Education Portal

Background: I own an expanding UK-based company which provides online English teaching services to companies and individuals in Russia.

Objective: To hire and properly monitor a team of English teachers in the UK, ensure student/teacher loyalty and cope with an expanding client base.

Maximum Load: 20 teachers, 200 students, 10 classes conducted simultaneously.

Current website: www.englishinrussia.ru

Site concept: Online education portal for Russian learners of English.

Competitors: <http://www.englishdom.com>, <http://www.english-natali.ru>

Features to be added:

- 1) **Stand-alone server software** in addition to the existing webserver.
 - Platform is subject to discussion
 - Its URL can be different from main website
 - Russian language support is a must
 - Database
- 2) **On-site web-conferencing system** similar to Skype, but simpler. This is an essential element of the site to guarantee control of the teaching process and to maintain teacher/student loyalty.
 - Ability to save information about classes: teacher, student, topic, start/end date and time, mark, comment
 - Two-way video stream
 - Quality audio
 - Text chat
 - Screen share
 - Whiteboard functionality
 - Option of recording lessons
- 3) **Back office for registered students.**
 - Login
 - Real-time updated student timetables
 - Teacher availability timetables
 - Payment system
 - Lesson booking and links to payment system
 - Teacher profiles with samples from lessons (embedded video, audio)
 - Internal messaging system for sending homework and questions/answers between students and teachers with the option of attaching files.
- 4) **Payment system.** Payment for lessons and courses online (similar to this: <http://www.englishdom.com/en/cost>).
- 5) **Online test functionality.** Interactive exercises in the form of multiple choice questions (similar to this: <http://cliomsk.com/on-line-test>). This should be created as a customisable template, which can be used to create other interactive exercises by study topic.
- 6) **Reports.**
 - Ability to monitor who is currently online, who is in the virtual classrooms.
 - Recorded lessons with start date/time and duration

-
- Payments list within a selected period, teacher or student.
 - Teacher workload statistics for every teacher
 - Report on student: classes with marks
 - Online test results.
 - Message logs.
 - System log.

7) **Configuration.**

- User page
- Student profiles
- Teacher profiles
- Online test templates
- Hourly rates for every teacher

Group 2

Mini Project: Software Testing and Quality Assurance Mini Project Dynamic website of covid-19 information using HTML, CSS, JAVASCRIPT And PHP, MySQL database used to store user account, comment, and registration form details. Regular Expression test cases for testing purpose.

OR

Mini Project: 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.

OR

Mini Project: 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.