DOCTOR PATIENT

Management system testing

**Submitted By: -**

# SIDDHIKA PAGARE TYCS-B 8323

Mahatma Education Society’s

**Pillai College of Arts, Commerce & Science**



**(Autonomous)**

**Affiliated to University of Mumbai**

NAAC Accredited 'A' grade (3 cycles)

Best College Award by University of Mumbai ISO 9001:2015 Certified

**CERTIFICATE**

*This is to certify that Mr. /Miss. Siddhika Pagare* *of*

***T.Y B.Sc. C.S. Semester VI*** *has completed the project work in the* *Subject of* ***Software Testing*** *during the academic year 2022-23 under the guidance of Prof****. Shubhangi Pawar*** *being* *the partial requirement for the fulfilment of the curriculum of*

***Degree of Bachelor of Science in Computer Science****,*

***University of Mumbai****.*

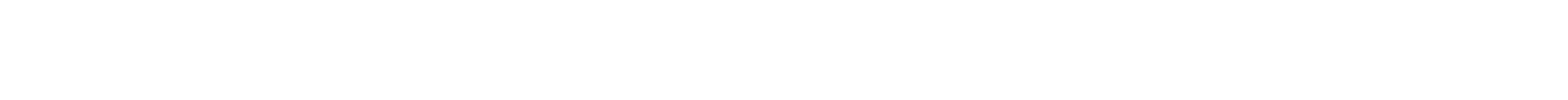
***Place:***

***Date:***

*Name & Signature of faculty Name & Signature of external*

*Name & Signature of Co-ordinator*

Dr. K.M. Vasudevan Pillai Campus, Sector 16, New Panvel - 410206.



Tel: 27456100 / 1700 | Fax: 27483208 | Website: [www.pcacs.ac.in](http://www.pcacs.ac.in/)

# INTRODUCTION

The "Doctor Patient Management System" has been developed to override the problems prevailing in the practicing manual system.

This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover,

this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while

entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it proves it is user-friendly.

Doctor Patient Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources.

## Software testing

• **Software Testing:**

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

### • Types of software testing

1. **Manual Testing:**

Manual testing is the process of testing the software by hand to learn more about it, to find what is and isn’t working. This usually includes verifying all the features specified in requirements documents, but often also includes the testers trying the software with the perspective of their end users in mind. Manual test plans vary from fully scripted test cases, giving testers detailed steps and expected results, through to high-level guides that steer exploratory testing sessions.

1. **Automation Testing:**

Automation testing is the process of using automation testing tools to control the execution of tests and compare the results against expected outcomes and to find the defects. In this process, testers execute the test scripts and generate the test results automatically by using automation tools.

Some of the famous automation testing tools for functional testing include Selenium, testRigor, and Katalon Studio.

Selenium is no longer a strange name for web application testers. It offers powerful capabilities like cross-browser testing but is difficult to learn for those new to automation or with limited programming experience.

**Manual testing using Excel:**

Excel provides extensive functionality for manipulating data. Customers can quickly generate values for tests with features like "fill series" and "import external data," so it is sensible to use Excel for writing functional tests. It is particularly appropriate to employ Excel when customers are testing complex business logic as the calculations can be modeled in the worksheet and then used to provide the expected results of a test.

Although you can save your Excel file in HTML format, the resulting file is not suitable for use as an input file for FIT, so some form of conversion is going to be required.

**Selenium:**

Selenium is an open-source, automated testing tool used to test web applications across multiple browsers. It's primarily built in Java and supports several browsers and programming languages.

* **The Selenium test suite comprises of four tools:**

i. Selenium Integrated Development Environment (IDE) ii. Selenium Remote Control (RC) iii. Selenium WebDriver iv. Selenium Grid

* + **Selenium IDE:**

Selenium IDE is part of the Selenium suite and was developed to speed up the creation of automation scripts. It’s a rapid prototyping tool and can be used by engineers with no programming knowledge whatsoever.

* + **Selenium WebDriver:**

Selenium WebDriver. Developed by Simon Stewart in 2006, Selenium WebDriver was the first cross-platform testing framework that could configure and control the browsers on the OS level. It served as a programming interface to create and run test cases.

* + **Selenium Remote Control:**

Selenium-Core was called "JavaScriptTestRunner," a tool built by

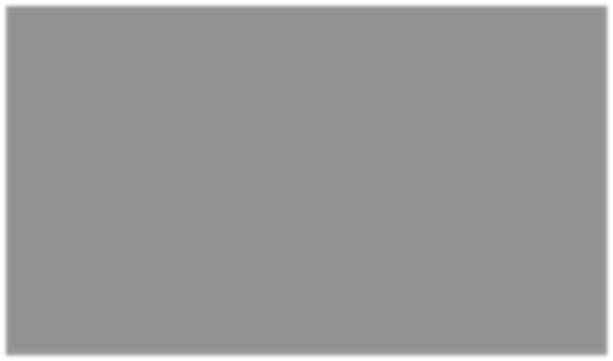
Jason Huggins in 2004. It was a set of JavaScript functions that interpreted and executed Selenium commands using the browser's built-in JavaScript interpreter. Selenium-Core was then injected into the web browser.

**Selenium Server Installation**

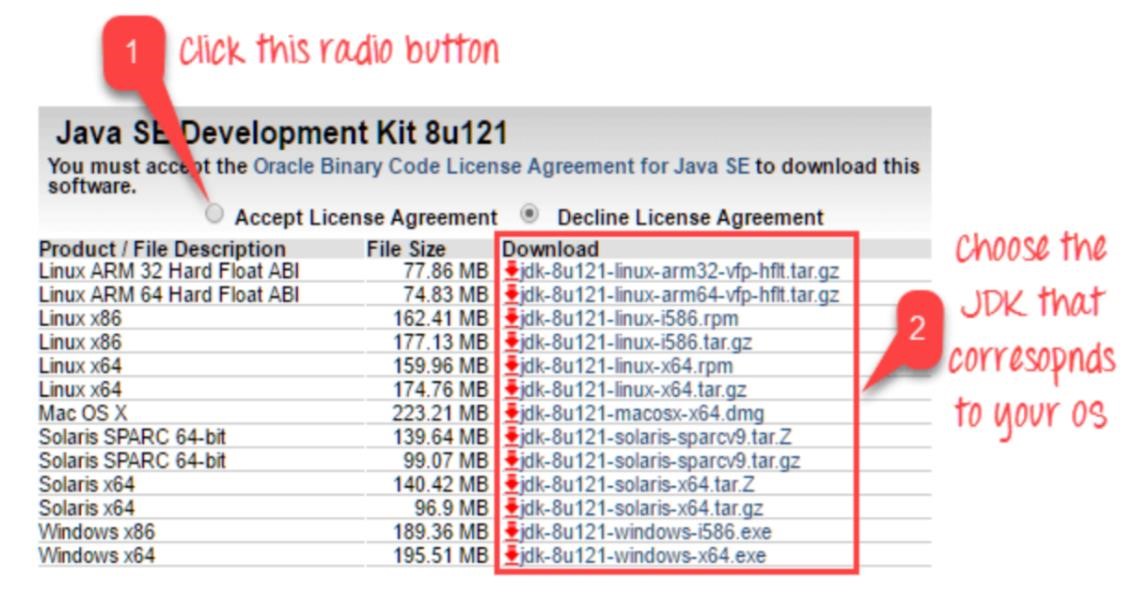
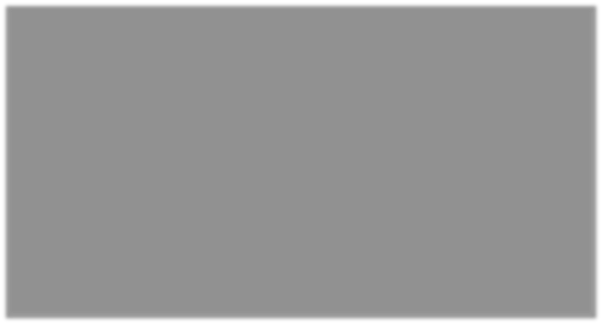
Selenium installation is a 3-step process:

* Install Java SDK
* Install Eclipse
* Install Selenium Driver Files

**Step 1** - Install Java on your computer



This JDK version comes bundled with Java Runtime Environment.



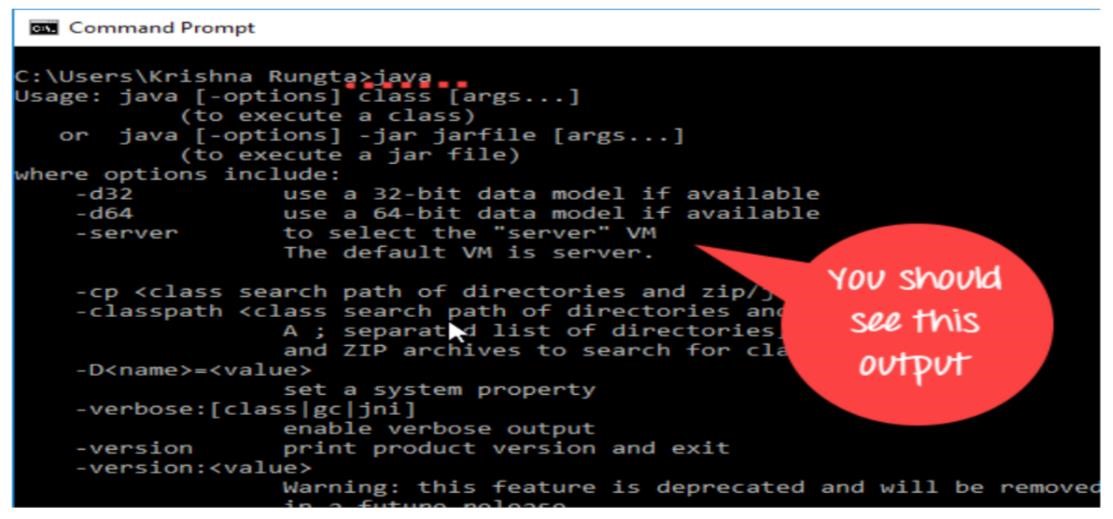
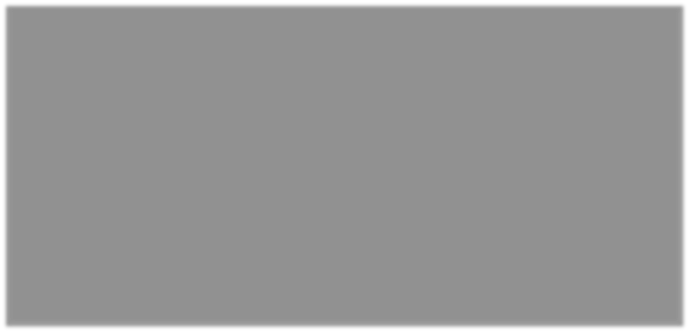
**Command prompt**

Once installation is complete, open command prompt and type “java”. If you see the following screen you are good to move to the next step

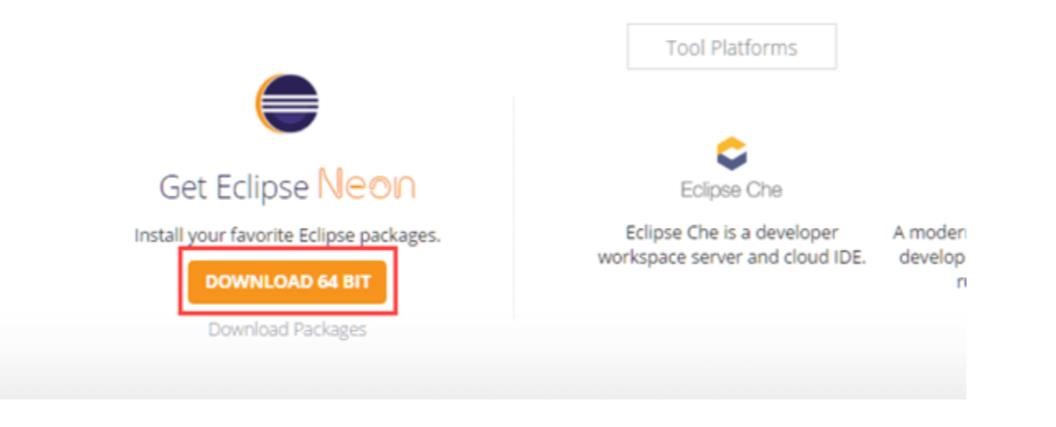
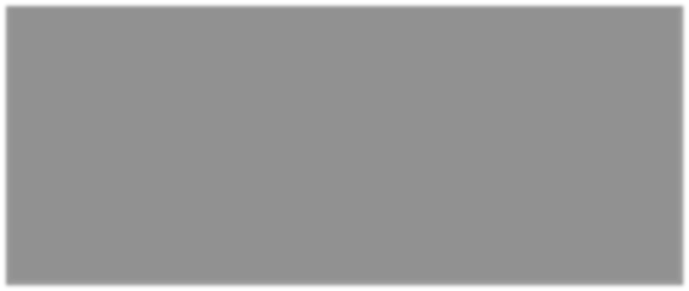
**Step 2**

-

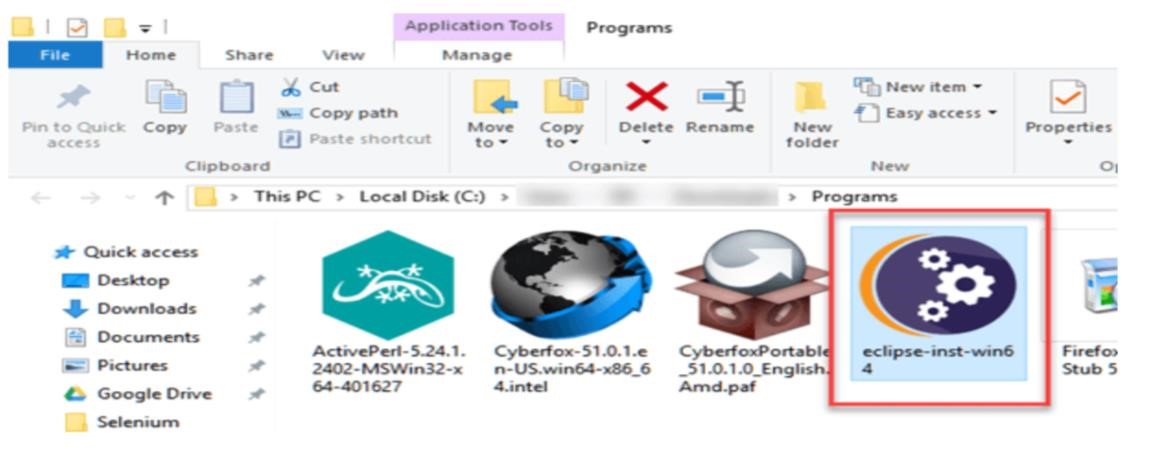
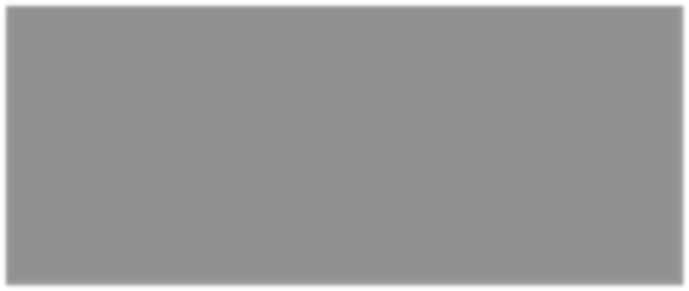
Install Eclipse IDE



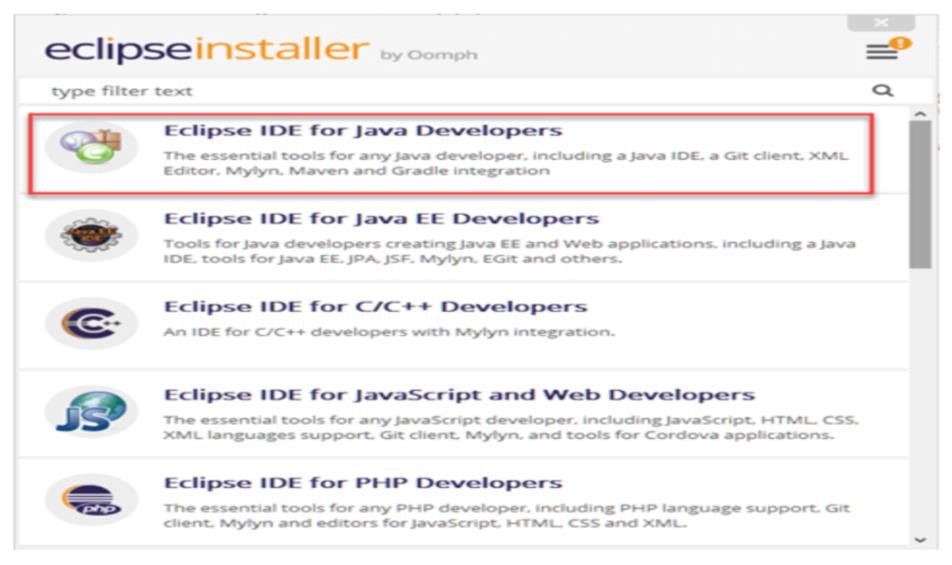
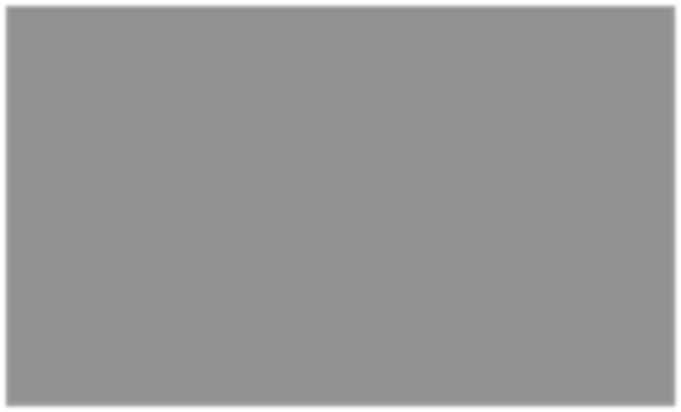
Download latest version of "Eclipse IDE for Java Developers" here. Be sure to choose correctly between Windows 32 Bit and 64 Bit versions.



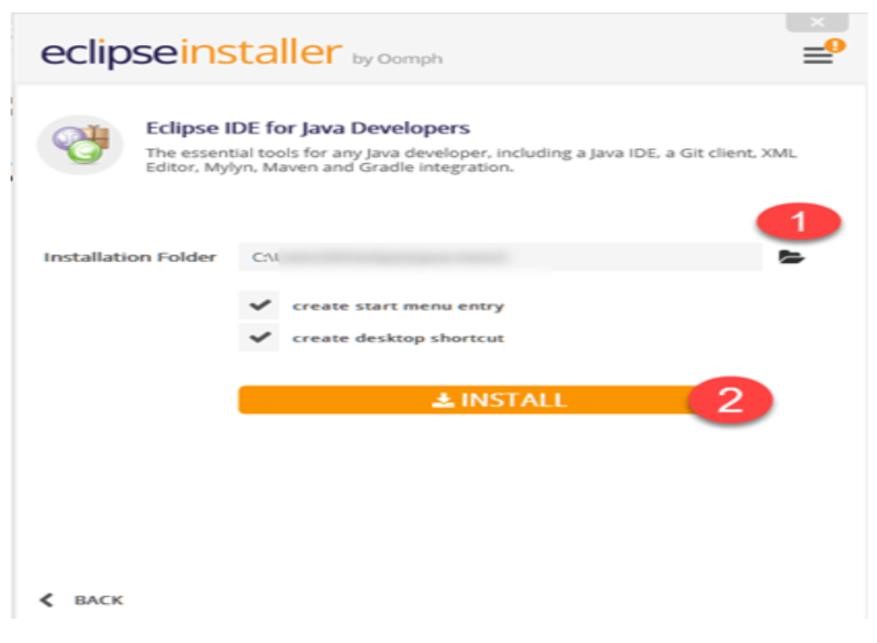
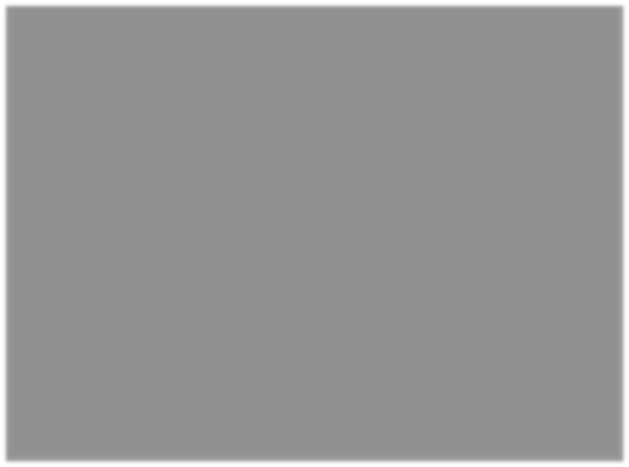
You should be able to download an exe file named "eclipse-inst-win64" for Setup.



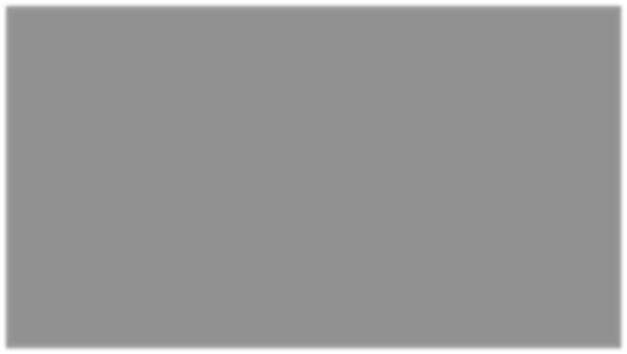
Double-click on file to Install the Eclipse. A new window will open. Click Eclipse IDE for Java Developers.



After that, a new window will open which click button marked 1 and change path to "C:\eclipse". Post that Click on Install button marked 2

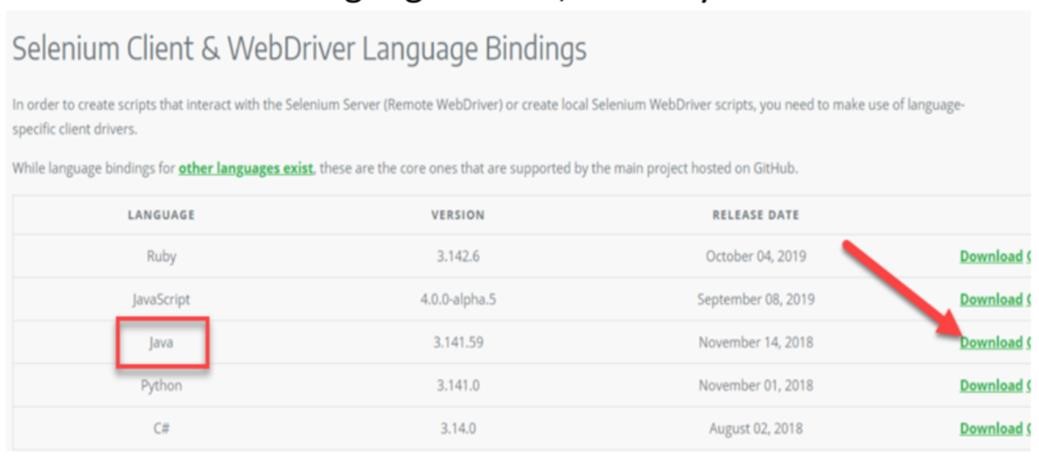
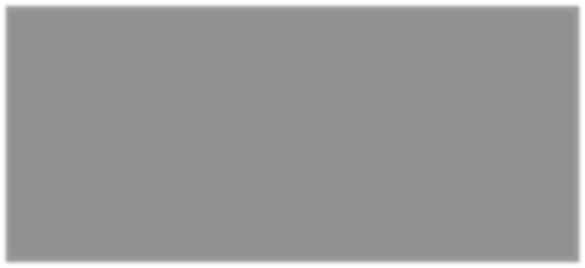


After successful completion of the installation procedure, a window will appear. On that window click on Launch



**Step 3** - Download the Selenium Java Client Driver.

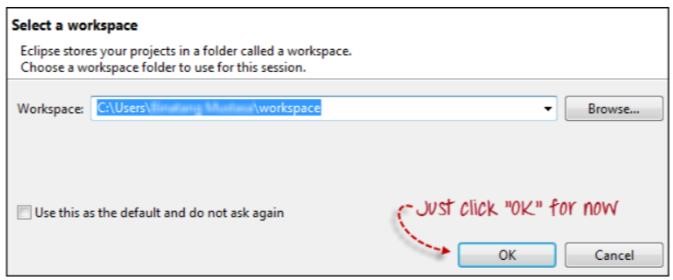
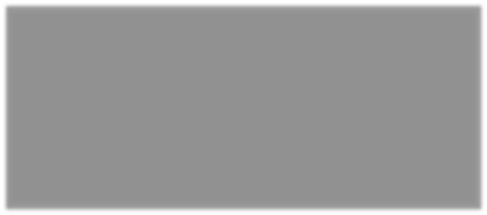
You can download the Selenium Java Client Driver here. You will find client drivers for other languages there, but only choose the one for Java.



This download comes as a ZIP file named "selenium-3.14.0.zip". For implicity, extract the contents of this ZIP file on your C drive so that you would have the directory "C:\selenium-3.14.0\". This directory contains all the JAR files that we would later import on Eclipse.

**Step 4** - Configure Eclipse IDE with WebDriver

* Launch the "eclipse.exe" file inside the "eclipse" folder that we extracted in step 2.
* If you followed step 2 correctly, the executable should be located on C:\eclipse\eclipse.exe. When asked to select for a workspace, just accept the default location.



Manual Testing

ADMIN LOGIN

Graphical user interface

Description automatically generated with low confidence

DOCTOR LOGIN

Graphical user interface, application

Description automatically generated

ADD/DELETE DOCTOR:

Graphical user interface, application

Description automatically generated with medium confidence

automation testing

To test number of links in the page

Code:

package Firefox;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class count\_item {

public static void main(String[] args) {

WebDriver driver=new ChromeDriver();

//driver.get("http://localhost/apr/APR/index.php");

//driver.manage().window().maximize();

driver.get("http://localhost/Doctor\_Patient/presentaionlayer/admin/viewdoctor.php");

java.util.List<WebElement> links = driver.findElements (By.tagName("a"));

System.out.println(" ");

System.out.println("Counting the total links on page");

System.out.println(" ");

System.out.println("Total links are"+links.size());

for (int i = 0; i<links.size(); i=i+1)

{

System.out.println("Link"+ i +"Link name "+links.get(i).getText());

}

}

}

Timeline

Description automatically generated with low confidence

Text

Description automatically generated

To test number of table rows in the page

Code:

package Firefox;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class count\_item {

public static void main(String[] args) {

WebDriver driver=new ChromeDriver();

driver.get("http://localhost/Doctor\_Patient/applicationlayer/login2.php");

driver.manage().window().maximize();

driver.findElement(By.name("doctorID")).sendKeys("1");

driver.findElement(By.name("doctorpassword")).sendKeys("123");

driver.findElement(By.name("Login2")).click();

//WebDriver driver=new ChromeDriver();

//driver.get("http://localhost/apr/APR/index.php");

//driver.manage().window().maximize();

driver.get("http://localhost/Doctor\_Patient/presentaionlayer/doctor/doctorapp.php");

java.util.List<WebElement> rows = driver.findElements (By.tagName("th"));

System.out.println(" ");

System.out.println("Counting the total rows on page");

System.out.println(" ");

System.out.println("Total rows are"+rows.size());

for (int i = 0; i<rows.size(); i=i+1)

{

System.out.println("Rows"+ i +"Rows name "+rows.get(i).getText());

}

}

}

Output:

Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated