



**Savitribai Phule Pune University**

**A**

**Project Report**

**On**

**“Regular Expression testcases for testing  
purpose”**

Submitted by

Mr. Swapnil Rajendra Take

UNDER THE GUIDANCE BY

**Prof. Pachhade R.C.**



**DEPARTMENT OF COMPUTER ENGINEERING**  
**VISHWABHARATI ACADEMY'S COLLEGE OF ENGINEERING SAROLA BADDI**  
**AHMEDNAGAR 414201**

**ACADEMIC YEAR 2022-23**



**DEPARTMENT OF COMPUTER ENGINEERING  
VISHWABHARATI ACADEMY'S COLLEGE OF ENGINEERING**

**Sarola Baddi, Ahmednagar**

**CERTIFICATE**

This is to certify that **Swapnil Rajendra Take** has successfully completed his Report on “Regular Expression testcases for testing purpose” at Vishwabharti Academy’s College of Engineering, Ahmednagar in the partial fulfillment of the Graduate Degree course in B.E. at the Department of Computer Engineering, in the academic Year 2022-2023

Semester-VII as prescribed by the Savitribai Phule Pune University

Prof. Pachhade R.C.

Project Guide

Prof. Joshi S.G.

Head of Department

Prof. Dhongde V.S.

Principal

Date:

Place: Ahmednagar

## **Acknowledgement**

We would like to extend our sincere appreciation and indebtedness to the teacher of the Computer Department Prof. Pachhade R.C. for providing the technical, informative support, valuable guidance and constant inspiration and encouragement as a project guide which has brought this stage one project report in this form.

We would also like to express our gratitude to Prof. Dhongade V.S. for his constant source of encouragement and friendly guidance throughout the project work And at the end we would like to express our gratitude to all staff member who have directly or indirectly contributed in their own way and all my friends Computer Department for their suggestions and constructive criticism.

Mr. Swapnil Rajendra Take

# **Table of Contents**

1. Introduction
2. Steps
3. Selenium Testing
4. Regular Expression
5. Sample Program
6. Screenshots
7. Test Results
8. Used Tools
9. Conclusion

## **Title:**

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 testcases for testing purpose

## **Problem Definition:**

Perform regular expression testing using selenium.

## **Objective**

We are going to learn how to Prepare Test Cases inclusive of Test Procedures for identified TestScenarios. Perform selenium testing with regular expression check  
Prepare Test Reports based on Test Pass/Fail Criteria.

## **Theory**

### **Test Plan for Website Testing**

The Test Plan document is derived from the Product Description, Software Requirement Specification, Use Case Documents. The focus of the test is what to test, how to test, when to test, and who will test. Test plan document is used as a communication medium between test team and test managers.

A standard test plan for Website Testing should define following features;

- Define the scope of testing
- Define objective of testing
- Approach for testing activity
- Schedule for testing
- Bug tracking and reporting

## **Steps for Download & Install Selenium WebDriver :**

1. Install Java on your computer
2. Install Eclipse IDE
3. Download the Selenium Java Client Driver
4. Configure Eclipse IDE with WebDriver

## **SELENIUM TESTING :**

Selenium is an open source umbrella project for a range of tools and libraries aimed at supporting browser automation. It provides a playback tool for authoring functional tests across most modern web browsers, without the need to learn a test scripting language. Regular expressions are a very useful technique for improving Selenium WebDriver tests. They can be used for

- extracting text from the value of a webelement
- validating if a value matches a specific pattern
- validating if a url matches a pattern

## **Validations Using Regular Expression :**

Let's look into regular expressions using a specific test case.  
Our test case does the following:

1. open the home page of a site (<http://www.vpl.ca>)
2. execute a keyword search (example: keyword = java)
3. the first results page is displayed
4. validate that the url is correct

ex. (<https://vpl.bibliocommons.com/search?q=java&t=keyword>)

5. validate that results count is greater than 0 (1 – 25 of 905 items; the results count is 905)
6. select page 2 of results
7. the second results page is displayed

8. validate that the url is correct

ex. ([https://vpl.bibliocommons.com/searchdisplay\\_quantity=25&page=2&q=java&t=keyword](https://vpl.bibliocommons.com/searchdisplay_quantity=25&page=2&q=java&t=keyword))

9. validate that the results count is greater than 0 (26 – 50 of 905 items; the results count is 905)

For the first validation, we should verify that the following urls are correct:

<https://vpl.bibliocommons.com/search?q=java&t=keyword>

[https://vpl.bibliocommons.com/search?display\\_quantity=25&page=2&q=java&t=keyword](https://vpl.bibliocommons.com/search?display_quantity=25&page=2&q=java&t=keyword)

**It would be great to use the same code to validate both urls since they are similar.**

The second url just adds 2 additional parameters.

For the **second validation**, we should extract the results count from the following texts:

1 – 25 of 905 items

26 – 50 of 905 items

Again, we should try to use the same code to extract the result count from both texts since they have the same pattern.

For both validations, we could use String methods and variables.

But regular expressions do the same things better.

First, some details about the project

The project uses 3 class :

1. Before class :

The @BeforeClass annotated method runs before the execution of test methods in a current class.

2. After class :

The @AfterClass annotated method will be executed after all the test methods of a current class have been invoked.

### 3. Test class :

The @Test class annotated method is containing test logic which is automate the webpage.

We are testing it on sign up page

#### **Code :**

```
package Test;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.AfterClass;

public class SignUpTest {
    public WebDriver driver;

    @Test
    public void Signup() {
        // For Open Website
        driver.get("http://localhost/Covid19-TMS%20Project%20Using%20PHP%20and%20MySQL/covid-tms/new-user-testing.php");

        // Sending full name
        driver.findElement(By.id("fullname")).sendKeys("Swapnil Rajendra Take");

        // Send phone to text field
        driver.findElement(By.id("mobilenumber")).sendKeys("8600789879");

        // dob
        driver.findElement(By.id("dob")).sendKeys("09-04-2000");
    }
}
```



```
// Govt Issued ID
driver.findElement(By.id("govtissuedid")).sendKeys("Adhar");

// ID Number
driver.findElement(By.id("govtidnumber")).sendKeys("979325686414");

// address
driver.findElement(By.id("address")).sendKeys("Newasa");

// State
driver.findElement(By.id("state")).sendKeys("Maharashtra");

//Test Type
driver.findElement(By.id("testtype")).sendKeys("RT-PCR");

//Time For test
driver.findElement(By.id("birthdaytime")).sendKeys("22-12-2022 13:20");

//Submit Button
driver.findElement(By.xpath("/html/body/div/div/div/div/form/div/div[2]/div/div[2]/div[3]/input")).click();

//String s=driver.getCurrentUrl();

}

@BeforeClass
public void beforeClass() { System.setProperty
("webdriver.chormedriver","C:\\Users\\Swapn\\OneDrive\\Desktop\\chromedriver.exe\\");

    driver = new ChromeDriver();

}

@AfterClass
public void afterClass() {

    driver.quit();

}

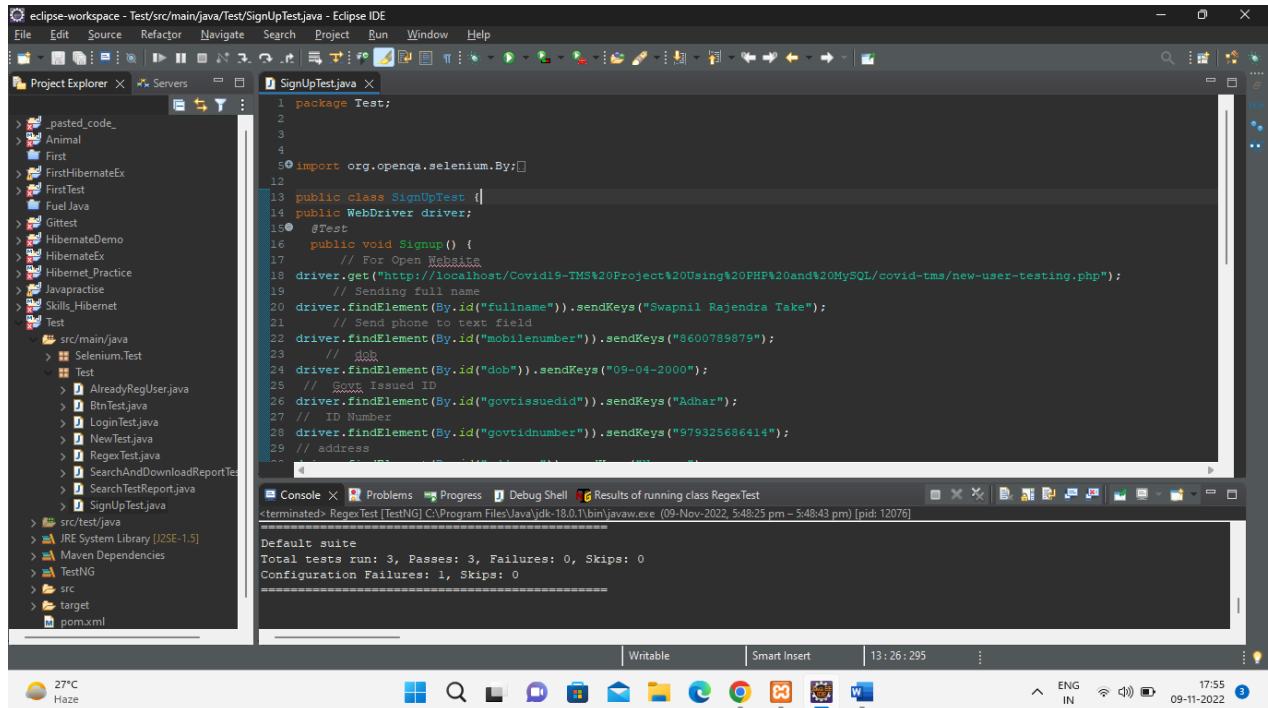
}
```

## **Test Scenarios for the Sign-up page :**

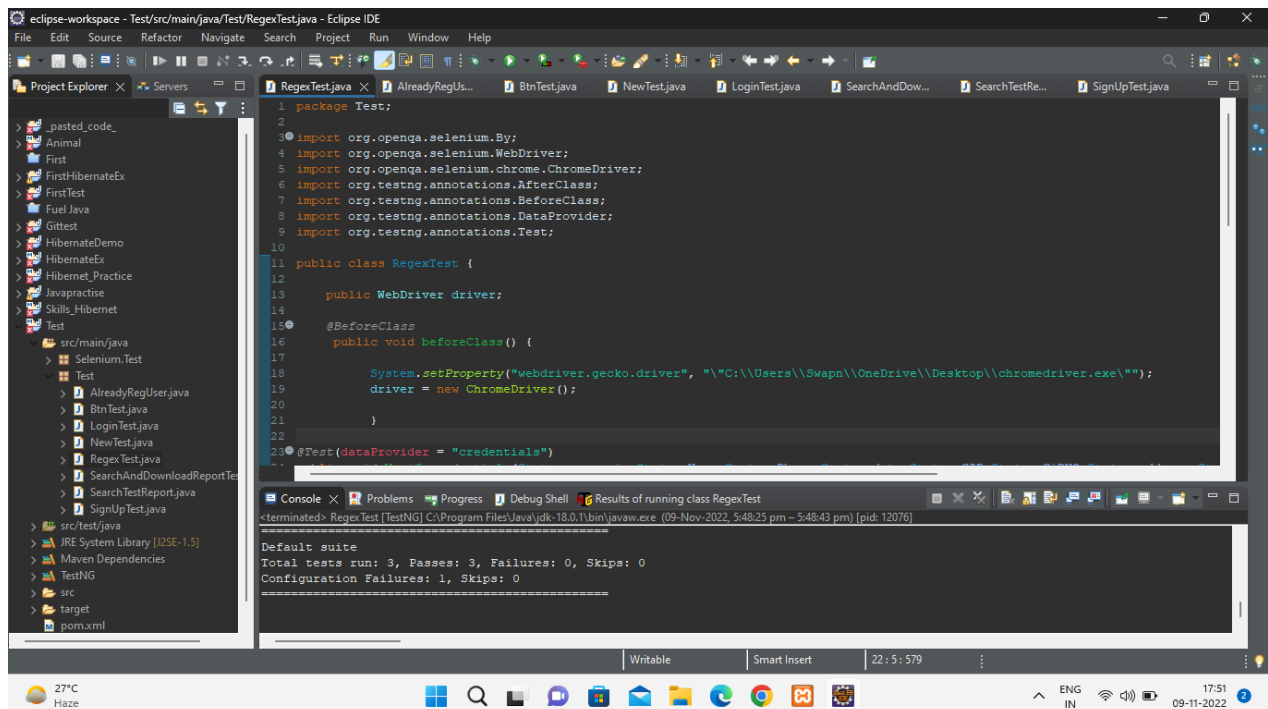
- 1) Verify the messages for each mandatory field.
- 2) Verify if the user cannot proceed without filling all the mandatory fields.
- 3) Verify the age of the user when the DOB is selected.
- 4) Verify if the numbers and special characters are not allowed in the First and Last name.
- 5) Verify if a 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.

# Screenshots :

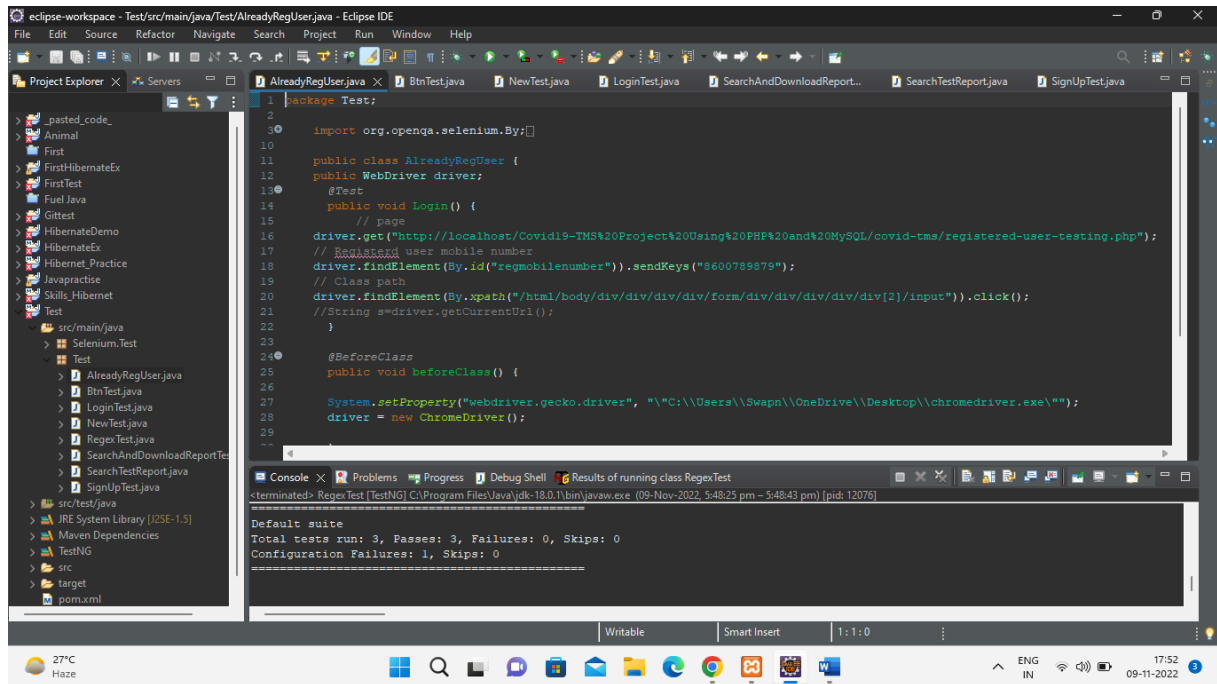
## 1. Signup page :



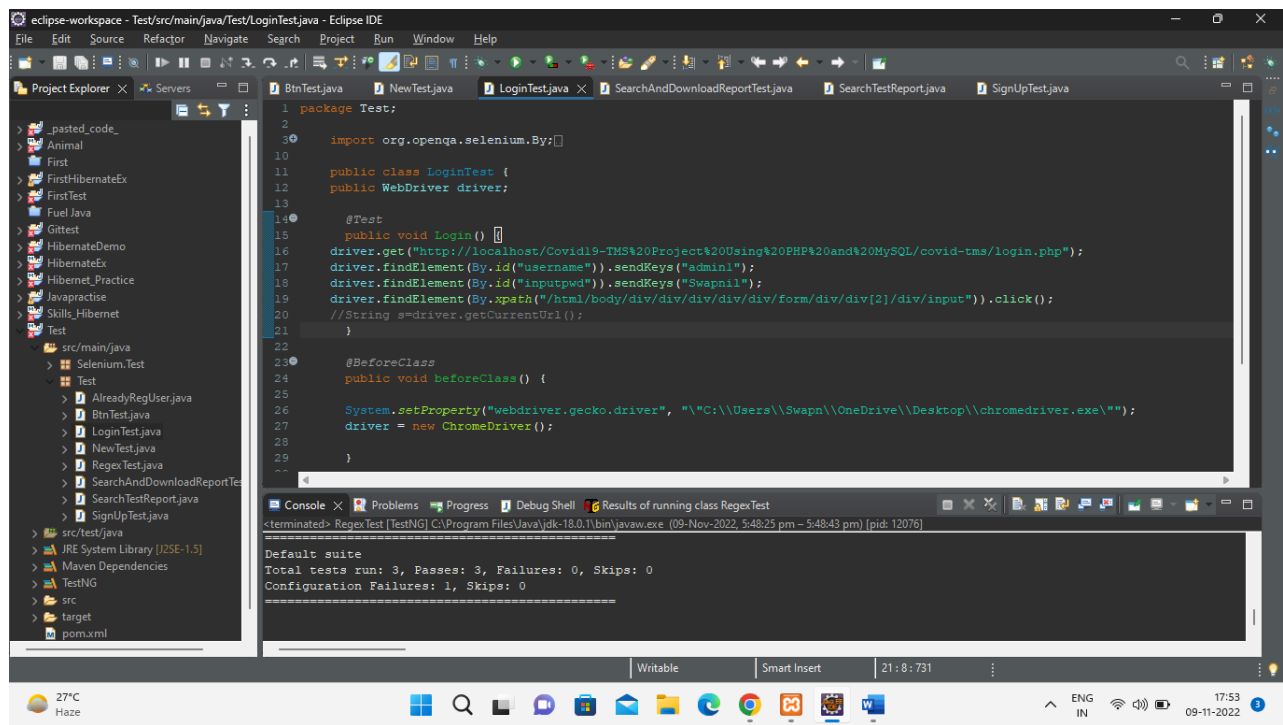
## 2. Regex test



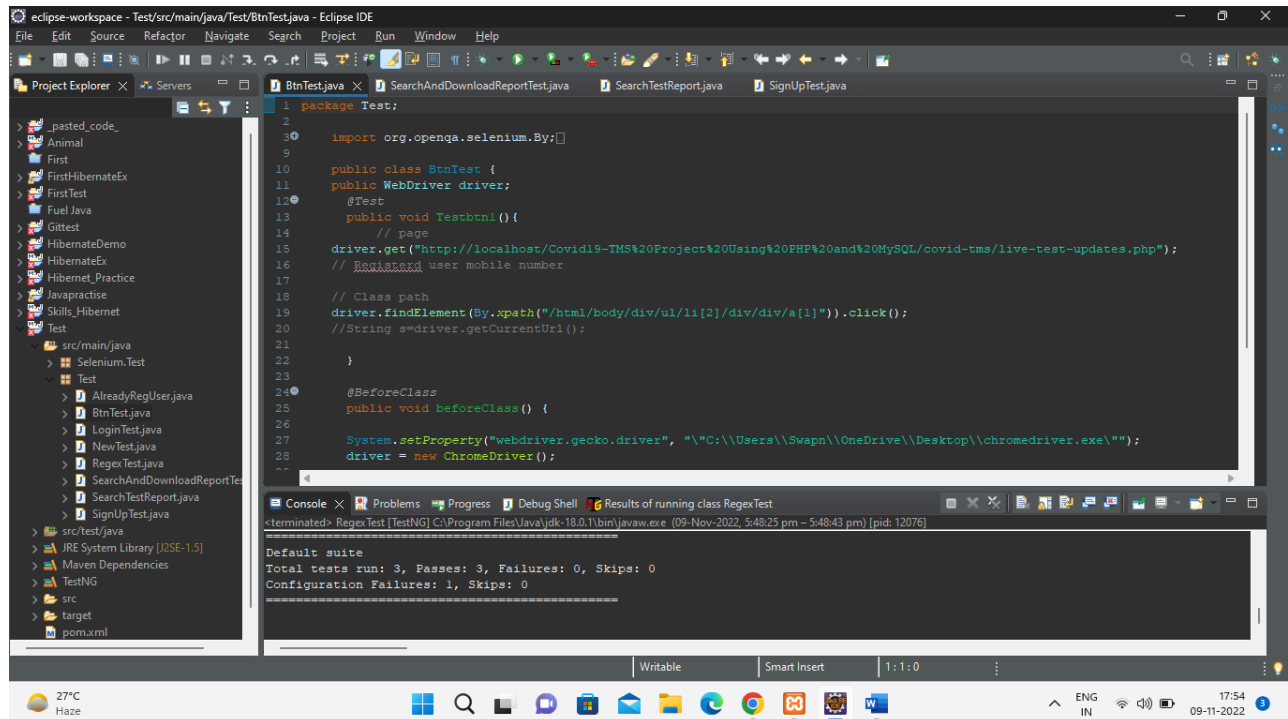
### 3. Registered user login :



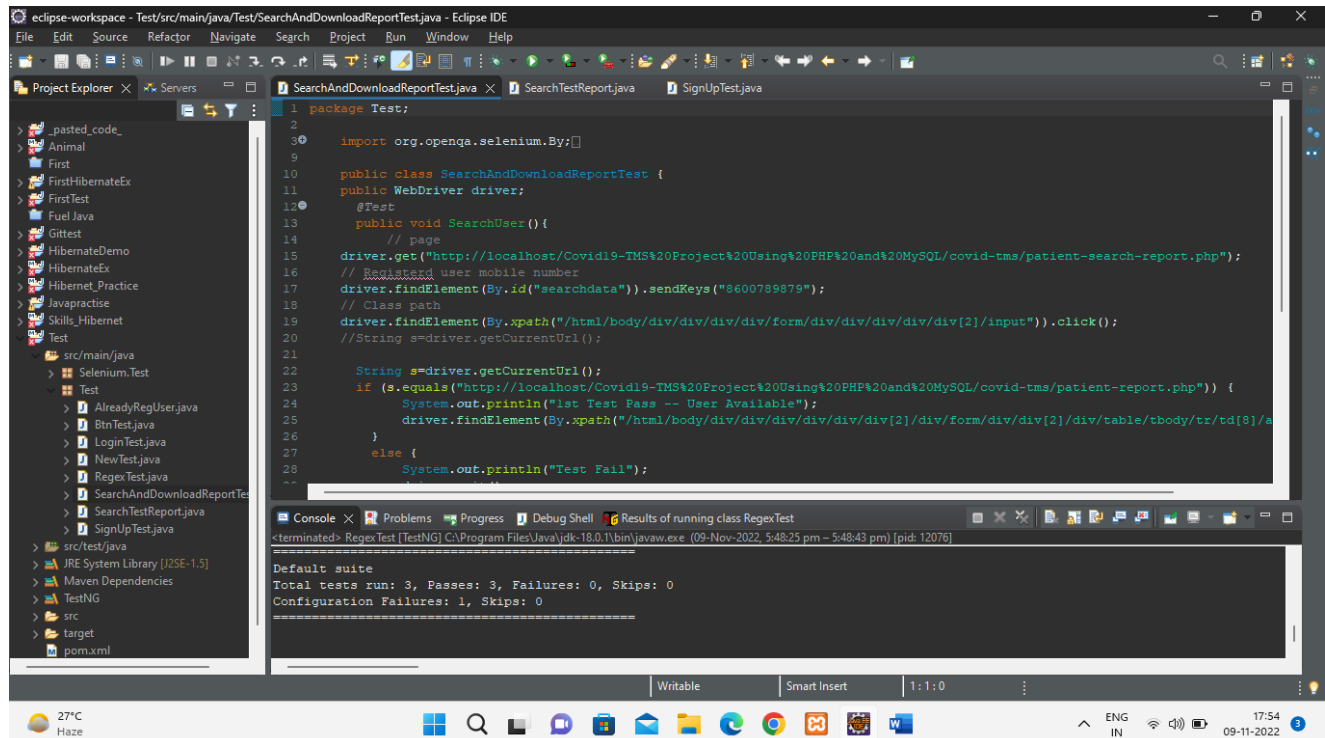
### 4. Login :



## 5. Button Test :



## 6. See Report :



# **Test Result Report**

## **TABLE OF CONTENTS**

1.0 Introduction

2.0 Testing Strategy

2.1 Selenium Testing

2.2 Regular Expression Testing

3.0 Tools

4.0 Approvals

## 1.0 INTRODUCTION

Project Dynamic webside of covid-19 information using HTML, CSS, JAVASCRIPT And PHP, MySQL database used to store user account, comment, and registration form details.

## 2.0 TESTING STRATEGY

1. **Functional testing**
2. **Selenium testing**

Approach	Type of Testing	Manual Testing		Automated Testing on Device	Tools/APIs/Libraries
		Using Device	Using Emulator		
Standard Testing (Functional Testing)	Selenium Testing	No	Yes	No	1. Selenium

### 2.1 Selenium Testing Report

EXECUTION STATUS	COMPLETED/CANCELLED/PENDING
PASSED TESTCASES	7
FAILED TESTCASES	2
PENDING TESTCASES	0
TEST CASES PLANNED	7

MODULES/ SCENARIOS	DESCRIPTION	% TCs EXECUTED	% TCs PASSED	TCs PENDING	PRIORITY	REMARKS
INSERT	Validation units related to insertion	63	37	00	MODERATE	-
VIEW	Retrieval of records from DB	100	100	00	LOW	-
SEARCH	Searching records in DB	100	100	00	MODERARTE	-

## 2.2 Regular Expression Testing

EXECUTION STATUS	COMPLETED/CANCELLED/PENDING
PASSED TESTCASES	3
FAILED TESTCASES	1
PENDING TERSTCASES	0
TEST CASES PLANNED	2

MODULES/ SCENARIOS	DESCRIPTION	% TCs EXECUTED	% TCs PASSED	TCs PENDING	PRIORITY	REMARKS
Email Validation	Email Id must have @ symbol.	100	100	00	HIGH	-
Phone No Validation	Phone no. must have 10 numbers.	100	100	00	HIGH	-
Patients Slot date validation	Patient's date must have next date from test apply.	100	100	00	HIGH	-



### **3.0 TOOLS**

1. Selenium jar file / Selenium IDE
2. Eclipse IDE
3. Testing extension in eclipse
4. Covid information site on localhost / website

### **Conclusion :**

Selenium is a cost-effective and flexible tool developers can use in the automation testing of their web applications. The most intriguing feature of this software is the ability to test applications across various web browsers. This ensures that websites do not crash or breakdown in certain browsers.

The above are examples of using regular expression in web tests. Regular expressions will be much more useful in API testing, where text-parsing is essential.

