Mini Project Report

ON

“TESTING OF NSS BLOG WEBSITE”

Submitted to

SAVITRIBAI PHULE PUNE UNIVERSITY

In partial fulfillment of Laboratory Practice-II

BY

Sushant Said (BECOMP-A54)
Jaid Mulani (BECOMP-A42)

UNDER THE GUIDANCE OF
Prof. Dnyaneshwar Choudhari

PIMPRI CHINCHWAD EDUCATION TRUST
A Trusted Brand in Education Since 1990..

Department of Computer Engineering
Pimpri Chinchwad College of Engineering and Research
Pune - 412101

[2020-2021]

PIMPRI CHINCHWAD EDUCATION TRUST’S

Pimpri Chinchwad College of Engineering and Research, Pune - 412101

CERTIFICATE

This is certify that the mini project report entitled

“NSS Blog Website Testing using Selenium”

submitted by

Mr. Sushant Said
Mr. Jaid Mulani

has successfully completed the mini project entitled “***NSS Blog Website Testing using Selenium***” in the fulfillment of B. E. (Computer Engineering) LP-II and this work has been carried out in my presence.

Date: / /
Place:

Prof. Dnyaneshwar Choudhari
Project Guide

Dr. Archana Chaugule
HOD
Department of Computer Engineering

Prof. Dr. Tiwari H.U.
Principal,PCCOE&R, Ravet

ACKNOWLEDGEMENT

It gives me great pleasure to present mini-project on **" NSS Blog Website Testing using Selenium"**. In preparing this report number of hands helped me directly and indirectly. Therefore, it becomes my duty to express my gratitude towards them.

I am very much obliged to subject guide **Asst Prof. Dnyaneshwar Choudhari** and Head of Computer department **Dr. Archana Chaugule** in Computer Engineering Department, for helping me and giving me proper guidance.

I am also thankful to my family for their whole hearted blessings are always for me support and constant encouragement towards the fulfillment of the work. I wish to record the help extended to be my friends in all possible ways and active support and constant encouragement.

Place: Ravet, Pune
Date: **Jaid Mulani (BECOMP-A42)**
Sushant Said (BECOMP-A54)

ABSTRACT

The website developed is an official website of National Service Scheme(NSS) unit at Pimpri Chinchwad College of Engineering and Research, Ravet. It has provision to provide a quick glimpse of the various activities conducted by NSS committee at PCCOER. It allows users to register themselves and create blogs and post photographs along with the description of the event. It also has provision to allow administrator to edit the contents of the blog and delete them. Activity gallery allows the admin to get an overview of the content posted in public domain on the website. Changes or updates to the website are near to real time and is backed by an efficient database.

Contents

List of Figures

1 Introduction

The website developed is an official website of National Service Scheme (NSS) unit at Pimpri Chinchwad College of Engineering and Research, Ravet. It has provision to provide a quick glimpse of the various activities conducted by NSS committee at PCCOER. Changes or updates to the website are near to real time and is backed by an efficient database

As this website represents the social participation of PCCOER in various activities inclined towards development of the society, it becomes extremely necessary to assure the acceptance and be sure that the website works well at all times. Hence, automation testing has been carried out using Selenium Webdriver. It is important to test the acceptance of the website before handing it over to the officials.

Creating a one stop platform that explains all the movements or activities of NSS was an important concern for PCCOER. Through collective effort it was possible to create a responsive and attractive website to fulfill the need. The website portrays every activity that is conducted by the NSS committee.

The website allows users to register themselves and create blogs and post photographs along with the description of the event. It also has provision to allow administrator to edit the contents of the blog and delete them. Activity gallery allows the admin to get an overview of the content posted in public domain on the website.

2 Problem Statement

Perform Web testing and identify the bugs using Selenium WebDriver and IDE and generate test reports encompassing exploratory testing. 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.

3 Testing

3.1 Test Objectives

The general objective of this study is to check the functionality of the web application properly. The specific objectives are as follows:

- Login as an admin to the NSS blog web application.
- Register as new user to the system to post new blogs.
- Post a new blog/article on website.
- Update the blog post.
- Delete a blog post.
- Check the functionality of Social Media handles of NSS.
- Test functionality of the entire web application.

3.2 Testing

Testing is the process of checking the functionality of an application to ensure it runs as per requirements. Unit testing comes into picture at the developers' level; it is the testing of single entity (class or method). Unit testing plays a critical role in helping a software company deliver quality products to its customers. Unit Testing can be done in two ways. Manual Testing and Automated Testing.

3.2.1 Manual Testing

Executing a test cases manually without any tool support is known as manual testing. Time-consuming and tedious, since test cases are executed by human resources, it is very slow and tedious. Huge investment in human resources as test cases need to be executed manually, more testers are required in manual testing. Manual testing is less reliable, as it has to account for human errors. No programming can be done to write sophisticated tests to fetch hidden information.

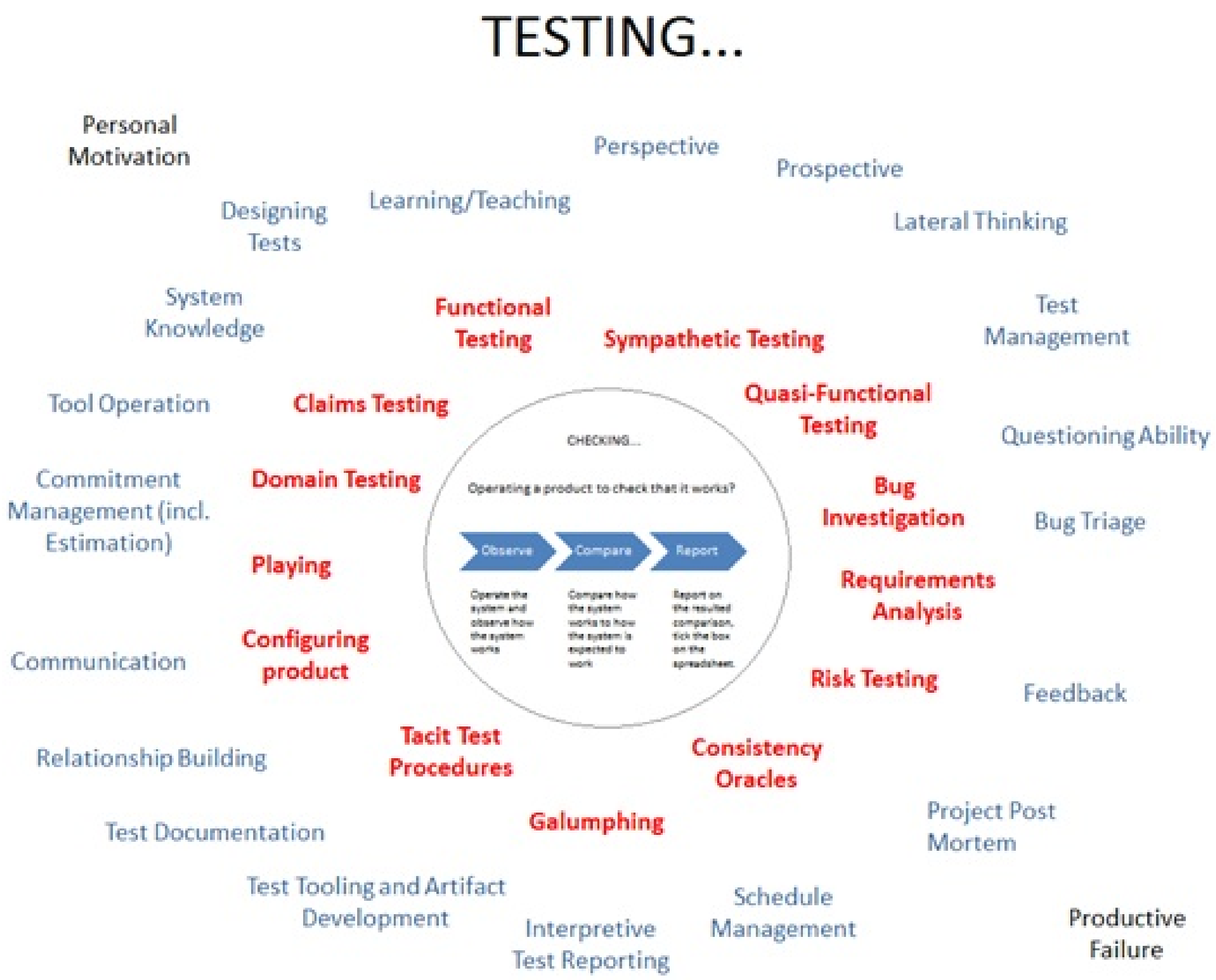


Figure 3.1: Testing Overview

3.2.2 Automated Testing

Taking tool support and executing the testcases by using an automation tool is known as automation testing. Fast automation runs test cases significantly faster than human resources. Test cases are executed using automation tools, so less number of testers are required in automation testing. More reliable Automation tests are precise and reliable. Programmable Testers can program sophisticated tests to bring out hidden information.

3.2.3 Alpha Testing

It is the most common type of testing used in the Software industry. The objective of this testing is to identify all possible issues or defects before releasing it into the market or to the user. Alpha Testing is carried out at the end of the software development phase but before the Beta Testing. Still, minor design changes may be made as a result of such testing. Alpha Testing is conducted at the developer's site. In-house virtual user environment can be created for this type of testing.

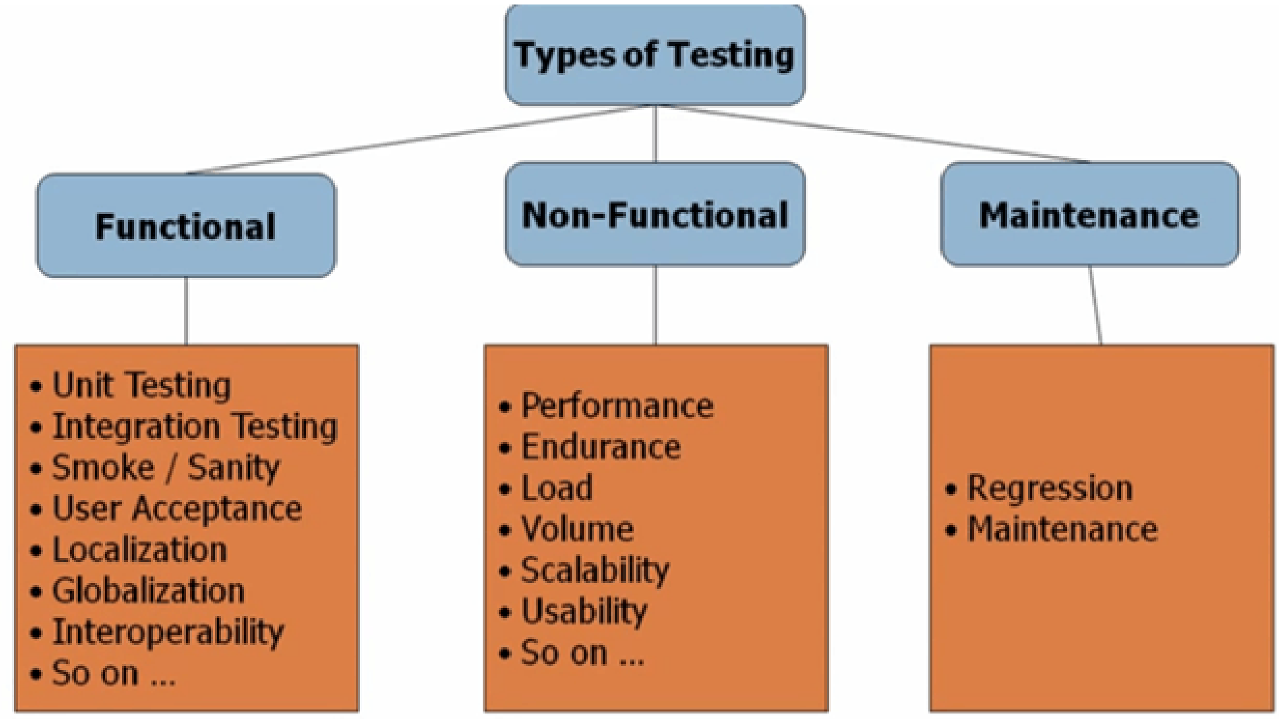
3.2.4 Acceptance Testing

An Acceptance Test is performed by the client and verifies whether the end to end the flow of the system is as per the business requirements or not and if it is as per the needs of the end-user. Client accepts the software only when all the features and functionalities work as expected.It is the last phase of the testing, after which the software goes into production. This is also called User Acceptance Testing (UAT).

3.2.5 Beta Testing

Beta Testing is a formal type of Software Testing which is carried out by the customer. It is performed in the Real Environment before releasing the product to the market for the actual end-users. Beta Testing is carried out to ensure that there are no major failures in the software or product and it satisfies the business requirements from an end-user perspective. Beta Testing is successful when the customer accepts the software. Usually, this testing is typically done by end-users or others. It is the final testing done before releasing an application for commercial purpose. Usually, the Beta version of the software or product released is limited to a certain number of users in a specific area.

Figure 3.2: Types of testing



3.3 Types of testing

Testing is an integral part of any successful software project. The types of software testing depend on various factors, including project requirements, budget, timeline, expertise, and suitability. The different Types of software testing are the key role where the tester determines the right testing for the apps. Functional testing and non- functional testing are the two kinds of testing performed by the QA or Software Tester.

These were some of the reasons that explain why we need software testing. Now let's move ahead with our Types of Software Testing blog and look at the different phases of Software Testing Life Cycle.

1. Functional Testing:
    - Unit Tesing
    - Integration Testing
    - Smoke / Sanity
    - User Acceptance
2. Non Functional Testing:
    - Performance
    - Usability
    - Volume
    - Scalability
    - Load
3. Maintenance :
    - Regression
    - Maintenance

4 Functional Requirements

In software engineering, a functional requirement defines a system or its component. It describes the functions a software must perform. A function is nothing but inputs, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.

Functional software requirements help you to capture the intended behavior of the system. This behavior may be expressed as functions, services or tasks or which system is required to perform.A requirement is a description of the service that software must offer. A requirement can range from the high-level abstract statement of the sender's necessity to detailed mathematical functional requirement specifications.

Benefits of Functional Requirement

- Helps you to check whether the application is providing all the functionalities that were mentioned in the functional requirement of that application.
- A functional requirement document helps you to define the functionality of a system or one of its subsystems.
- Functional requirements along with requirement analysis help identify missing requirements.
- They help clearly define the expected system service and behavior.
- Errors caught in the Functional requirement gathering stage are the cheapest to fix.
- Support user goals, tasks, or activities.

5 Non Functional Requirements

A non-functional requirement defines the quality attribute of a software system. They represent a set of standards used to judge the specific operation of a system. Example, how fast does the website load?

A non-functional requirement is essential to ensure the usability and effectiveness of the entire software system. Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.

Non-functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are ¿ 10000. Description of non-functional requirements is just as critical as a functional requirement.

Benefits of Non-Functional Requirement

- The non-functional requirements ensure the software system follow legal and compliance rules.
- They ensure the reliability, availability, and performance of the software system.
- They ensure good user experience and ease of operating the software.
- They help in formulating security policy of the software system.
- Non Functional Requirements offer key design specifications which support the architectural decisions made and in many ways can be seen as building blocks on which the system scalability depends.
- It also quantify what the investment is going to deliver and how far into the future can the application be expected to go in terms of meeting growth in end user workload.

6 Block Diagram

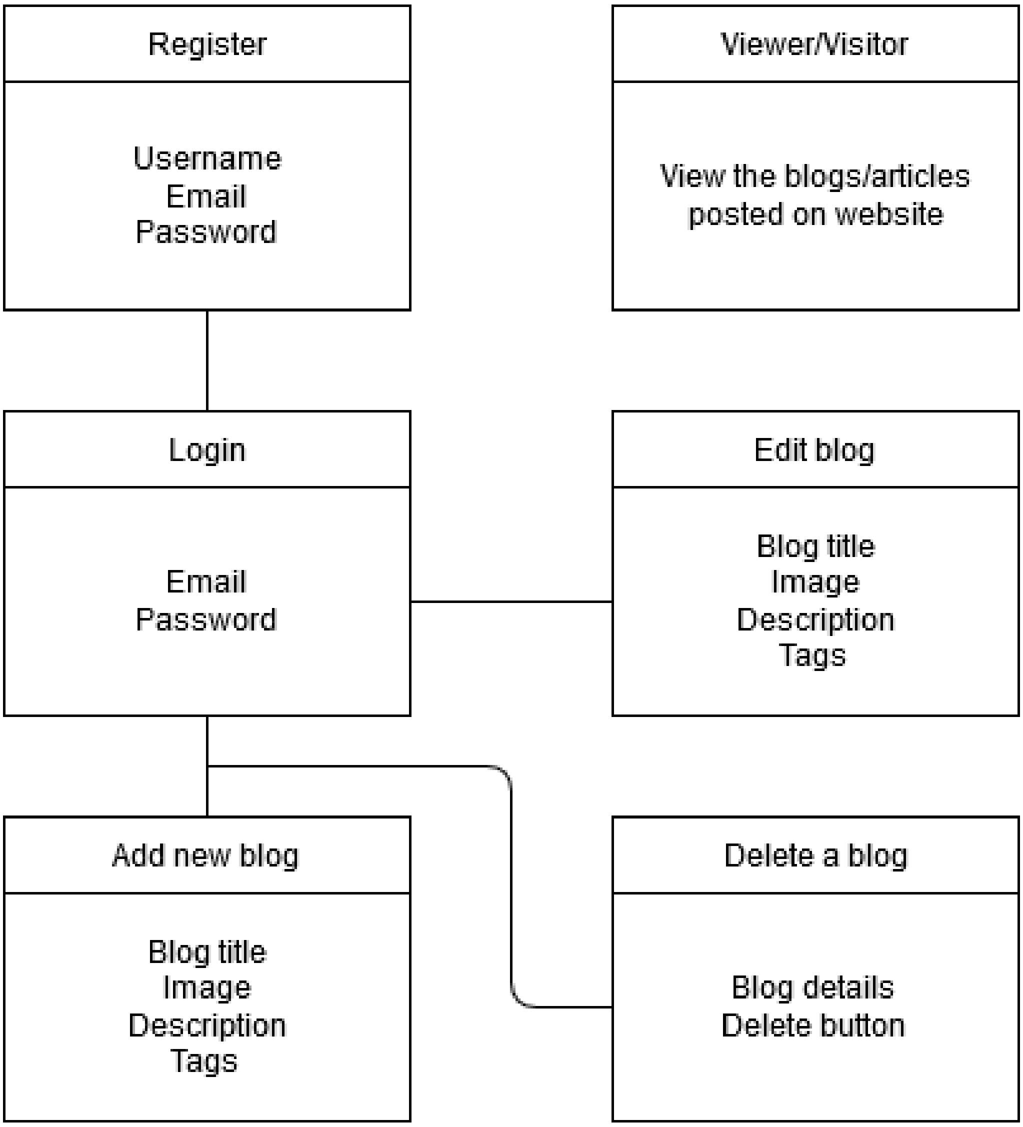


Figure 6.1: Block Diagram For NSS Blog Web application

7 Source Code / Functions

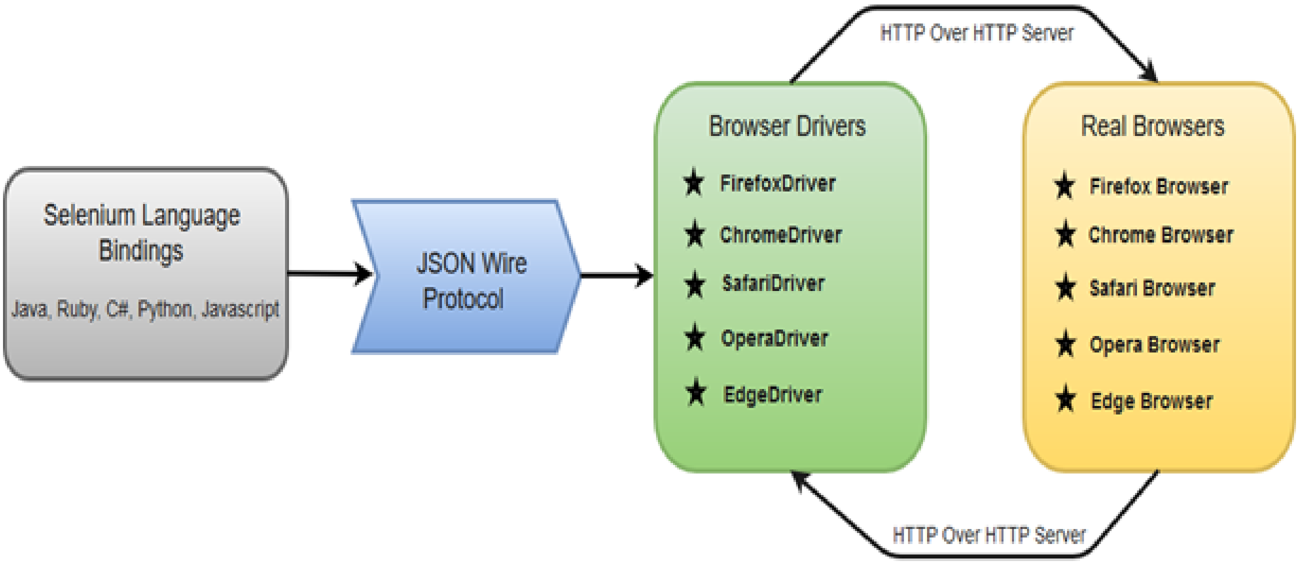
- Builder().forBrowser('chrome').build() : To let selenium webdriver automate browser operation.
- driver.get(): To get url of the web page
- driver.getAllWindowHandles(): To get information about all the open tabs in browser window.
- driver.getTitle(): To get the title of current page
- driver.findElement(): The first element with the name, value or id attribute value matching the location will be returned.
- element.click() : To make a click action on a component
- element.sendKeys("parameters") : To send parameters to a component
- assert.equal() : Assert function is used to make an assertion and check whether testcase has passed or failed.
- driver.quit(): To close the browser after all testcases are executed

8 Selenium WebDriver

8.1 Selenium WebDriver- Architecture

Selenium WebDriver API provides communication facility between languages and browsers. The following image shows the architectural representation of Selenium WebDriver. Selenium WebDriver fits in the same role as RC did, and has incorporated the original 1.x bindings. It refers to both the language bindings and the implementations of the individual browser controlling code. This is commonly referred to as just ”WebDriver” or sometimes as Selenium 2.

Figure 8.1: Selenium WebDriver- Architecture



- Selenium uses drivers, specific to each browser in order to establish a secure connection with the browser without revealing the internal logic of browser’s functionality. The browser driver is also specific to the language used for automation such as Python, Java, C, etc.
- JSON (JavaScript Object Notation) is an open standard for exchanging data on

web. It supports data structures like object and array. So, it is easy to write and read data from JSON.

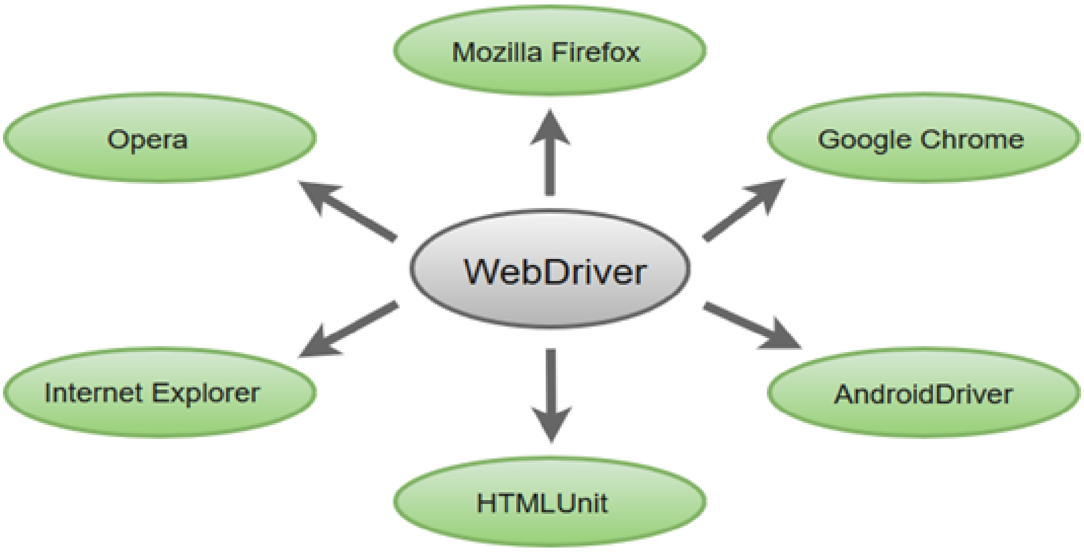
- Selenium developers have built language bindings/Selenium Client Libraries in order to support multiple languages. For instance, if you want to use the browser driver in Python, use the Python bindings.
- Selenium WebDriver performs much faster as compared to Selenium RC because it makes direct calls to the web browsers. RC on the other hand needs an RC server to interact with the browser.
- In WebDriver, test scripts can be developed using any of the supported programming languages and can be run directly in most modern web browsers. Languages supported by WebDriver include C, Java, Perl, PHP, Python and Ruby.
- Selenium WebDriver was first introduced as a part of Selenium v2.0. The initial version of Selenium i.e Selenium v1 consisted of only IDE, RC and Grid. However, with the release of Selenium v3, RC has been deprecated and moved to legacy package.
- Selenium WebDriver is the most important component of Selenium Tool's Suite. The latest release ”Selenium 2.0” is integrated with WebDriver API which provides a simpler and more concise programming interface.

8.2 Selenium WebDriver- Features

- Multiple Languages Support: WebDriver also supports most of the commonly used programming languages like Java, C, JavaScript, PHP, Ruby, Pearl and Python. Thus, the user can choose any one of the supported programming language based on his/her competency and start building the test scripts.
- Multiple Browser Support: Selenium WebDriver supports a diverse range of web browsers such as Firefox, Chrome, Internet Explorer, Opera and many more. It also supports some of the non-conventional or rare browsers like HTMLUnit.
- Speed: WebDriver performs faster as compared to other tools of Selenium Suite. Unlike RC, it doesn't require any intermediate server to communicate with the browser; rather the tool directly communicates with the browser.

- Simple Commands: Most of the commands used in Selenium WebDriver are easy to implement. For instance, to launch a browser in WebDriver following commands are used:
- WebDriver driver = new FirefoxDriver(); (Firefox browser)
- WebDriver driver = new ChromeDriver(); (Chrome browser)
- WebDriver- Methods and Classes: WebDriver provides multiple solutions to cope with some potential challenges in automation testing. WebDriver also allows testers to deal with complex types of web elements such as checkboxes, dropdowns and alerts through dynamic finders.
- There needs to be some actual signal in our features so that models built using those features do better than random guessing.
- The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other.

Figure 8.2: Selenium Features



9 Output

9.1 Pass/Fail criteria

This section specifies generic pass/fail criteria for the tests covered in this plan. They are supplemented by pass/fail criteria in the test design specification.

9.1.1 Component Pass/Fail criteria

- **Login module**

  It checks whether the website redirects the user to logged in home page after entering the necessary login credentials.

- **Register new user**

  The new user registration which requires user to create their username, password, add email id is tested.

- **Create new blog post module**

  On clicking the Add Article button it check whether the user is able to post new blog post on the web application.

- **Find the added Blog Post**

  New blog post details are entered and submit button is clicked, then we try to find the blog post, if details are fetched then testcase is satisfied.

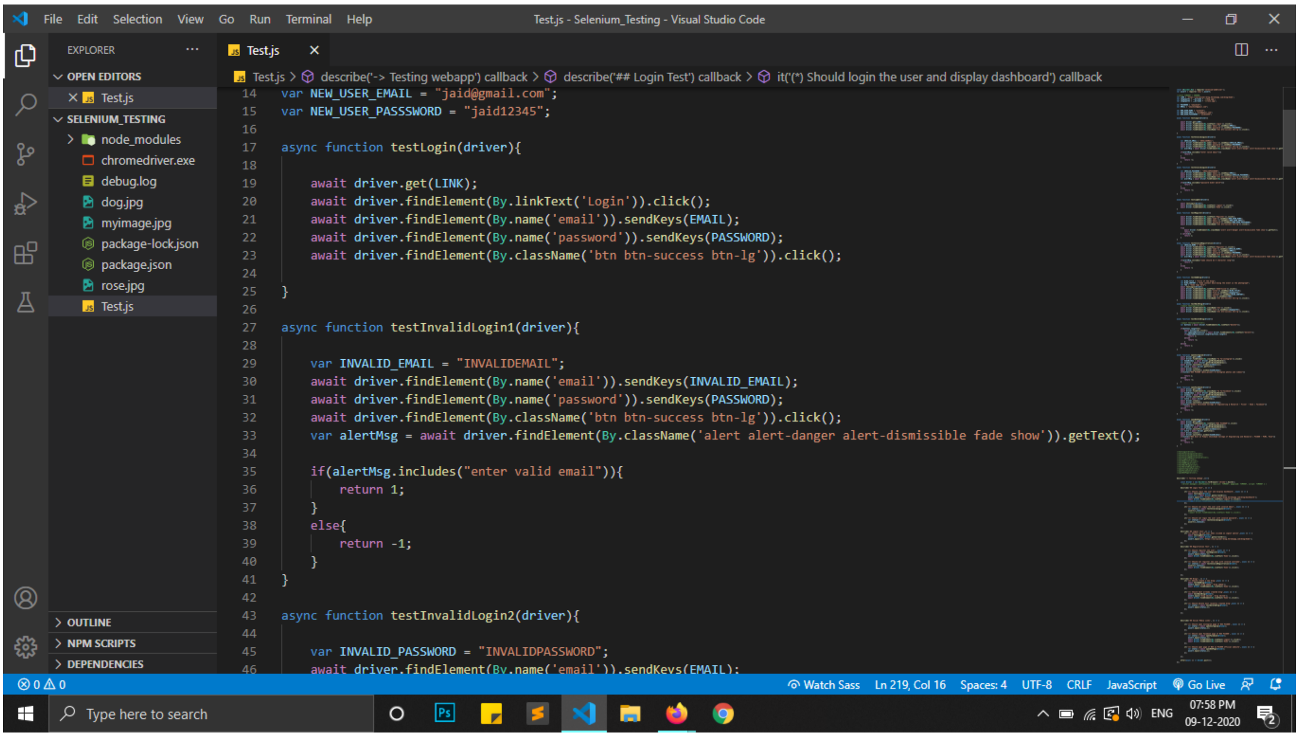
- **Updating details of existing blog or article**

  The details of an existing blog post are updated by clicking edit button and changing the details is verified.

- **Deleting a blog post**

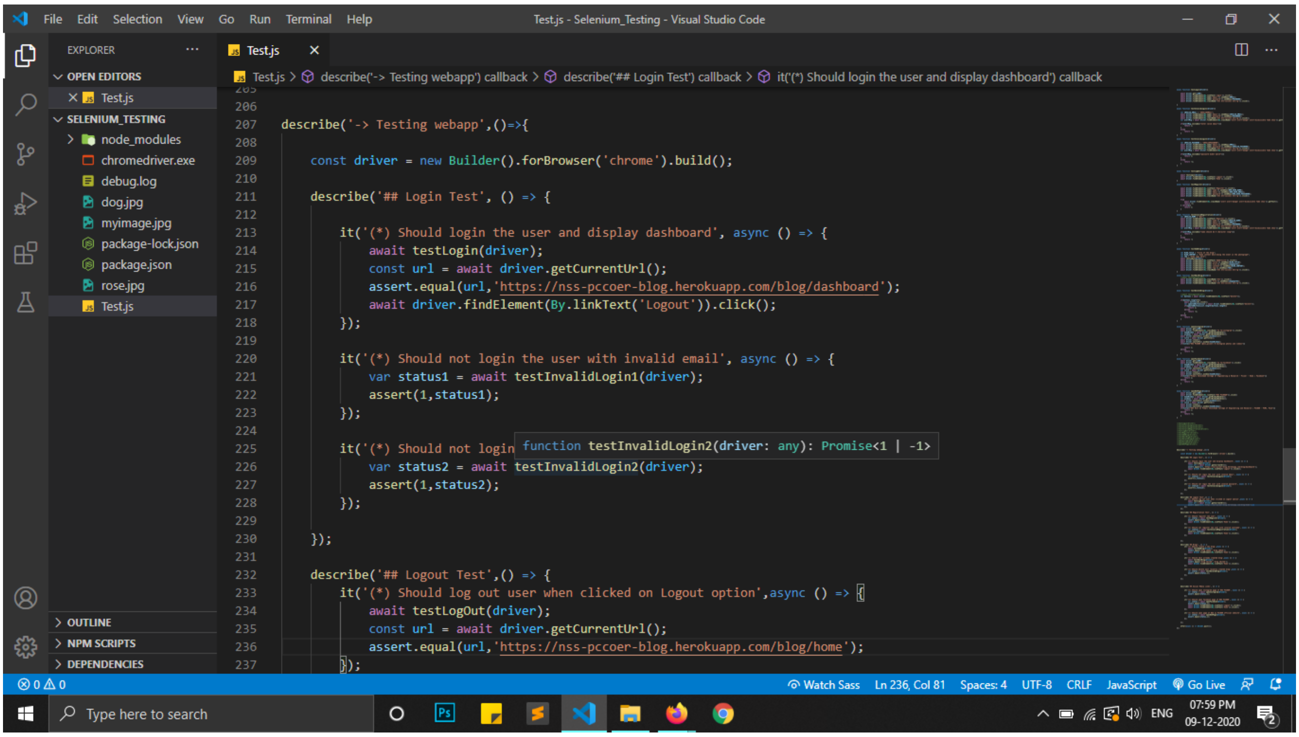
  The ID of the blog post is fetched from the database and using delete button user should be able to delete the post. This is tested.

10 Screenshots


File Edit Selection View Go Run Terminal Help
Test.js - Selenium_Testing - Visual Studio Code
EXPLORER
OPEN EDITORS
Test.js
SELENIUM_TESTING
node_modules
chromedriver.exe
debug.log
dog.jpg
myimage.jpg
package-lock.json
package.json
rose.jpg
Test.js
OUTLINE
NPM SCRIPTS
DEPENDENCIES
Test.js > describe('-> Testing webapp') callback > describe('## Login Test') callback > it('(*) Should login the user and display dashboard') callback
Watch Sass Ln 219, Col 16 Spaces: 4 UTF-8 CRLF JavaScript Go Live
Type here to search
ENG
07:58 PM
09-12-2020

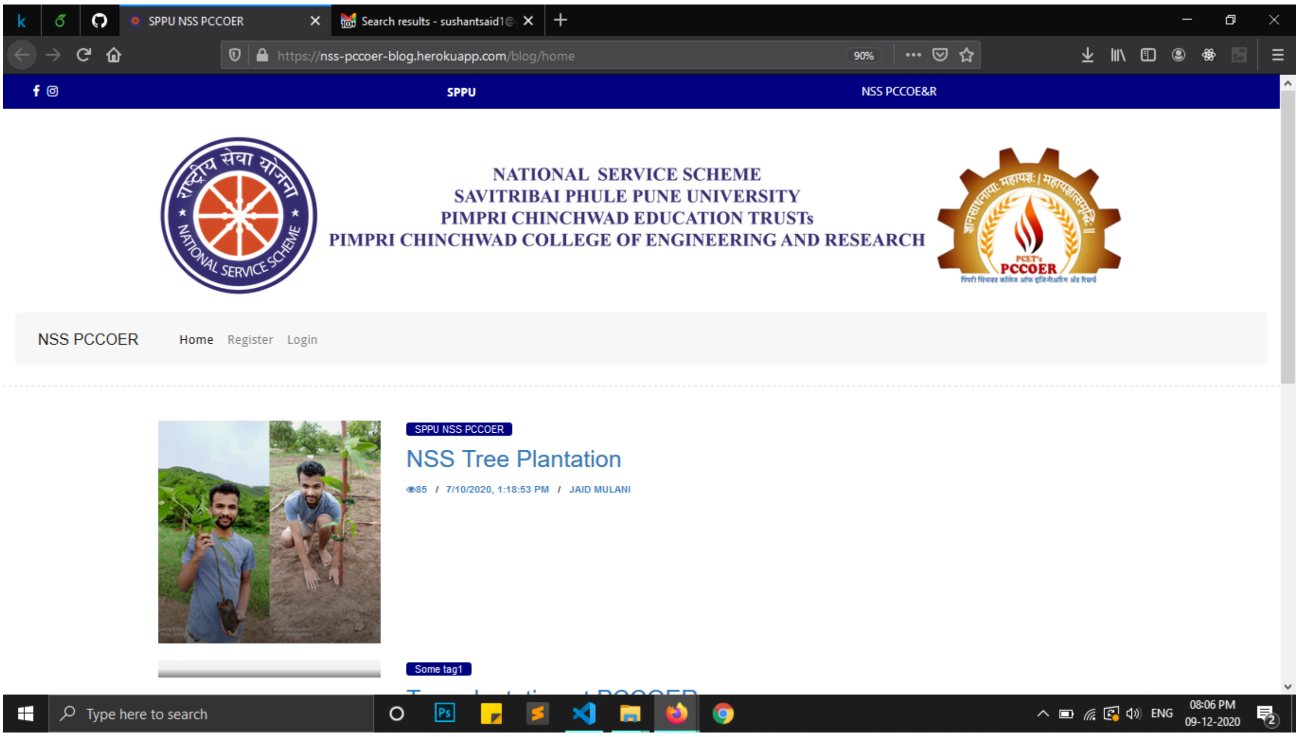
```javascript
var NEW_USER_EMAIL = "jaid@gmail.com";
var NEW_USER_PASSSWORD = "jaid12345";

async function testLogin(driver){

    await driver.get(LINK);
    await driver.findElement(By.linkText('Login')).click();
    await driver.findElement(By.name('email')).sendKeys(EMAIL);
    await driver.findElement(By.name('password')).sendKeys(PASSWORD);
    await driver.findElement(By.className('btn btn-success btn-lg')).click();

}

async function testInvalidLogin1(driver){

    var INVALID_EMAIL = "INVALIDEMAIL";
    await driver.findElement(By.name('email')).sendKeys(INVALID_EMAIL);
    await driver.findElement(By.name('password')).sendKeys(PASSWORD);
    await driver.findElement(By.className('btn btn-success btn-lg')).click();
    var alertMsg = await driver.findElement(By.className('alert alert-danger alert-dismissible fade show')).getText();

    if(alertMsg.includes("enter valid email")){
        return 1;
    }
    else{
        return -1;
    }
}

async function testInvalidLogin2(driver){

    var INVALID_PASSWORD = "INVALIDPASSWORD";
    await driver.findElement(By.name('email')).sendKeys(EMAIL);
```

Figure 10.1: Selenium Code



```
describe('-> Testing webapp',()=>{

    const driver = new Builder().forBrowser('chrome').build();

    describe('## Login Test', () => {

        it('(*) Should login the user and display dashboard', async () => {
            await testLogin(driver);
            const url = await driver.getCurrentUrl();
            assert.equal(url,'https://nss-pccoer-blog.herokuapp.com/blog/dashboard');
            await driver.findElement(By.linkText('Logout')).click();
        });

        it('(*) Should not login the user with invalid email', async () => {
            var status1 = await testInvalidLogin1(driver);
            assert(1,status1);
        });

        it('(*) Should not login function testInvalidLogin2(driver: any): Promise<1 | -1>
            var status2 = await testInvalidLogin2(driver);
            assert(1,status2);
        });

    });

    describe('## Logout Test',() => {
        it('(*) Should log out user when clicked on Logout option',async () => {
            await testLogOut(driver);
            const url = await driver.getCurrentUrl();
            assert.equal(url,'https://nss-pccoer-blog.herokuapp.com/blog/home');
        });
```

Figure 10.2: Selenium Code (Continued)



SPPU NSS PCCOER

https://nss-pccoer-blog.herokuapp.com/blog/home

SPPU NSS PCCOE&R

NATIONAL SERVICE SCHEME
SAVITRIBAI PHULE PUNE UNIVERSITY
PIMPRI CHINCHWAD EDUCATION TRUSTs
PIMPRI CHINCHWAD COLLEGE OF ENGINEERING AND RESEARCH

NSS PCCOER Home Register Login

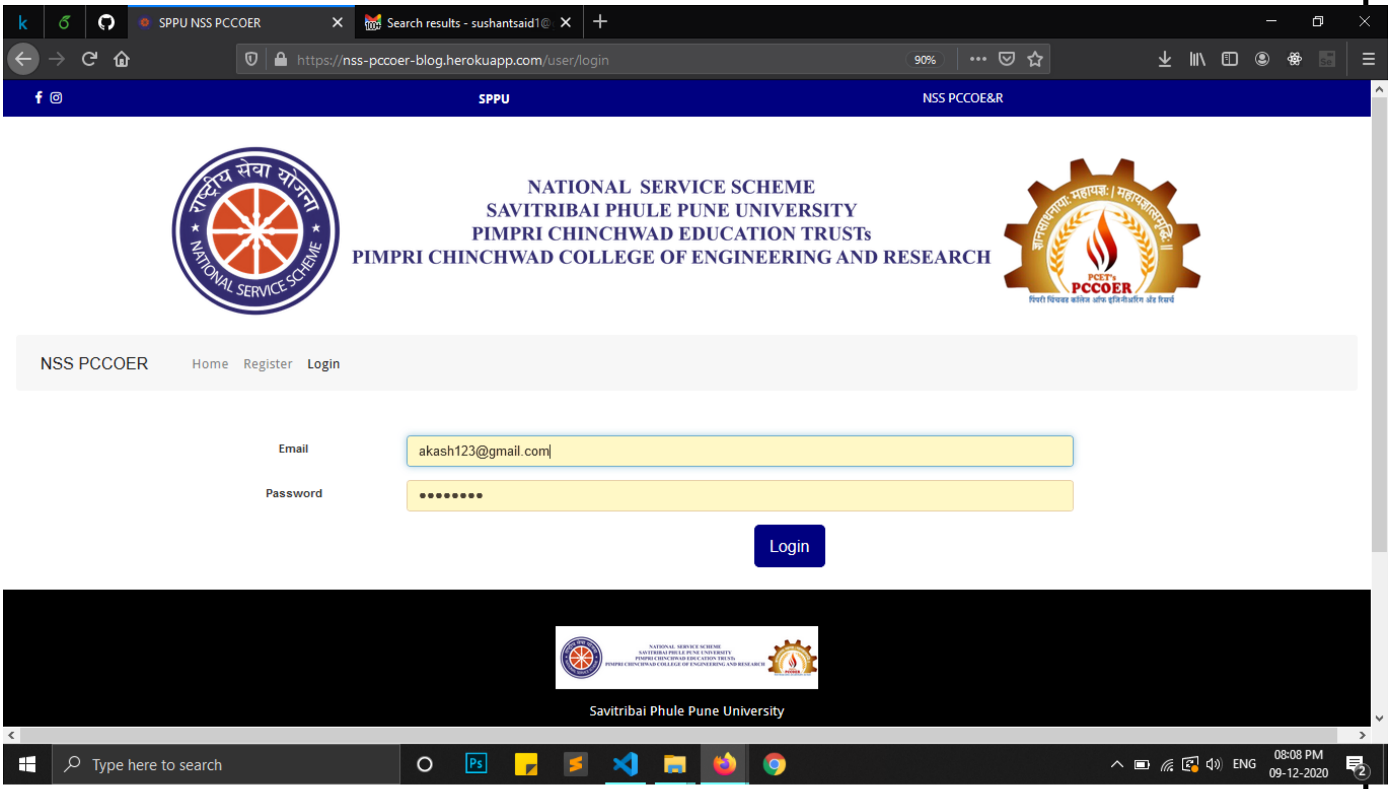
SPPU NSS PCCOER

NSS Tree Plantation

85 / 7/10/2020, 1:18:53 PM / JAID MULANI

Some tag1

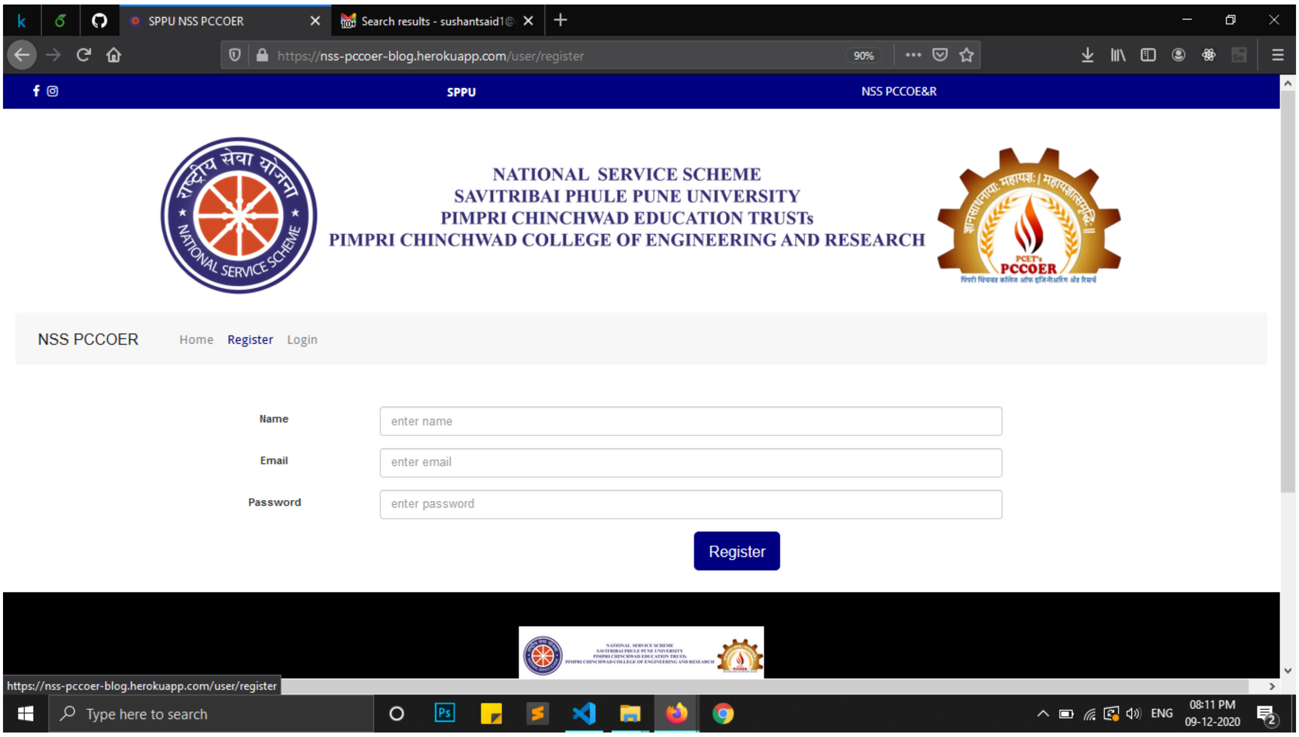
Figure 10.3: Website - Home Page


SPPU NSS PCCOER
Search results - sushantsaid1@
https://nss-pccoer-blog.herokuapp.com/user/login
90%
SPPU
NSS PCCOE&R
NATIONAL SERVICE SCHEME
SAVITRIBAI PHULE PUNE UNIVERSITY
PIMPRI CHINCHWAD EDUCATION TRUSTs
PIMPRI CHINCHWAD COLLEGE OF ENGINEERING AND RESEARCH
NSS PCCOER
Home
Register
Login
Login
Savitribai Phule Pune University
Type here to search
ENG
08:08 PM
09-12-2020

Email akash123@gmail.com

Password ••••••••

Figure 10.4: Login

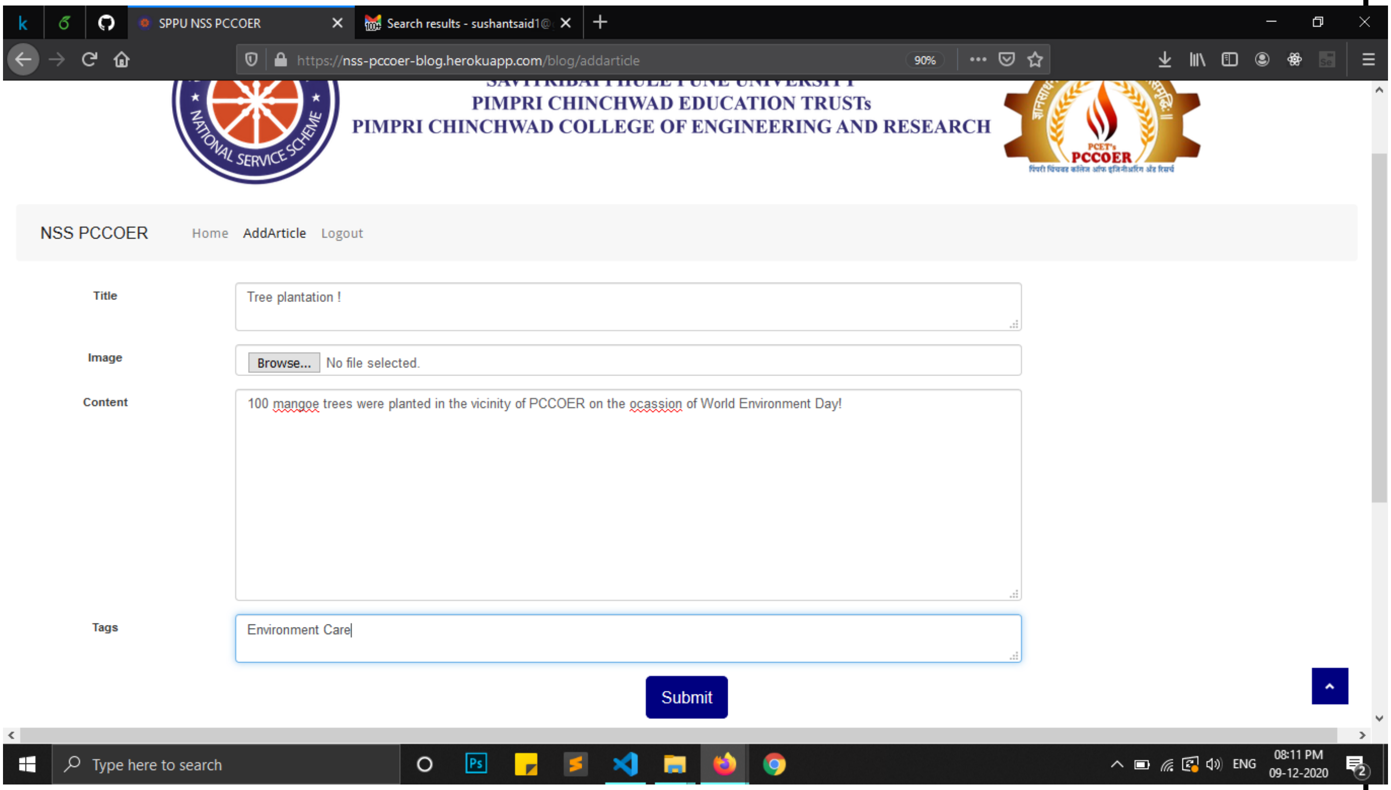

SPPU NSS PCCOER
Search results - sushantsaid1@
https://nss-pccoer-blog.herokuapp.com/user/register
90%
SPPU
NSS PCCOE&R
NATIONAL SERVICE SCHEME
SAVITRIBAI PHULE PUNE UNIVERSITY
PIMPRI CHINCHWAD EDUCATION TRUSTs
PIMPRI CHINCHWAD COLLEGE OF ENGINEERING AND RESEARCH
NSS PCCOER
Home
Register
Login
Register
https://nss-pccoer-blog.herokuapp.com/user/register
Type here to search
ENG
08:11 PM
09-12-2020

Name enter name

Email enter email

Password enter password

Figure 10.5: Register New User


SPPU NSS PCCOER
Search results - sushantsaid1@
https://nss-pccoer-blog.herokuapp.com/blog/addarticle
90%
SAVITRIBAI PHULE PUNE UNIVERSITY
PIMPRI CHINCHWAD EDUCATION TRUSTs
PIMPRI CHINCHWAD COLLEGE OF ENGINEERING AND RESEARCH
NSS PCCOER
Home
AddArticle
Logout
Type here to search
08:11 PM
09-12-2020

Title: Tree plantation !

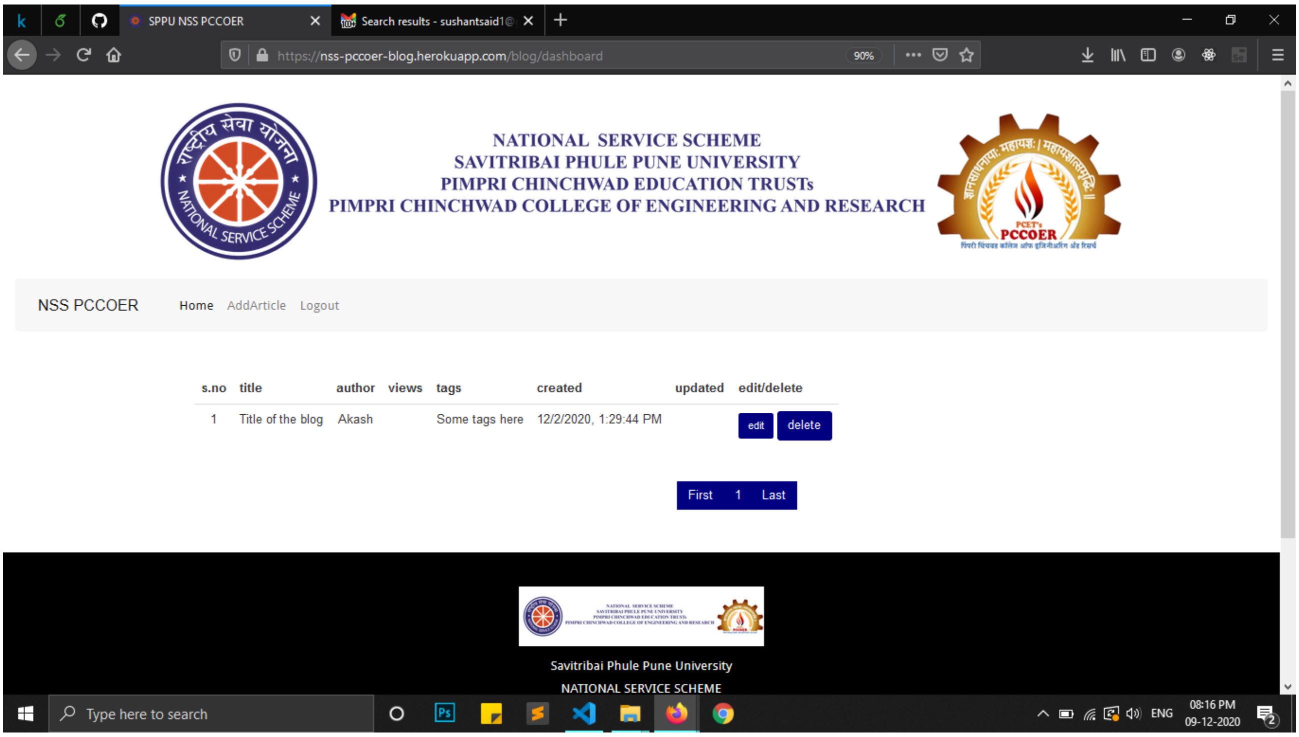
Image: Browse... No file selected.

Content: 100 mangoe trees were planted in the vicinity of PCCOER on the ocassion of World Environment Day!

Tags: Environment Care

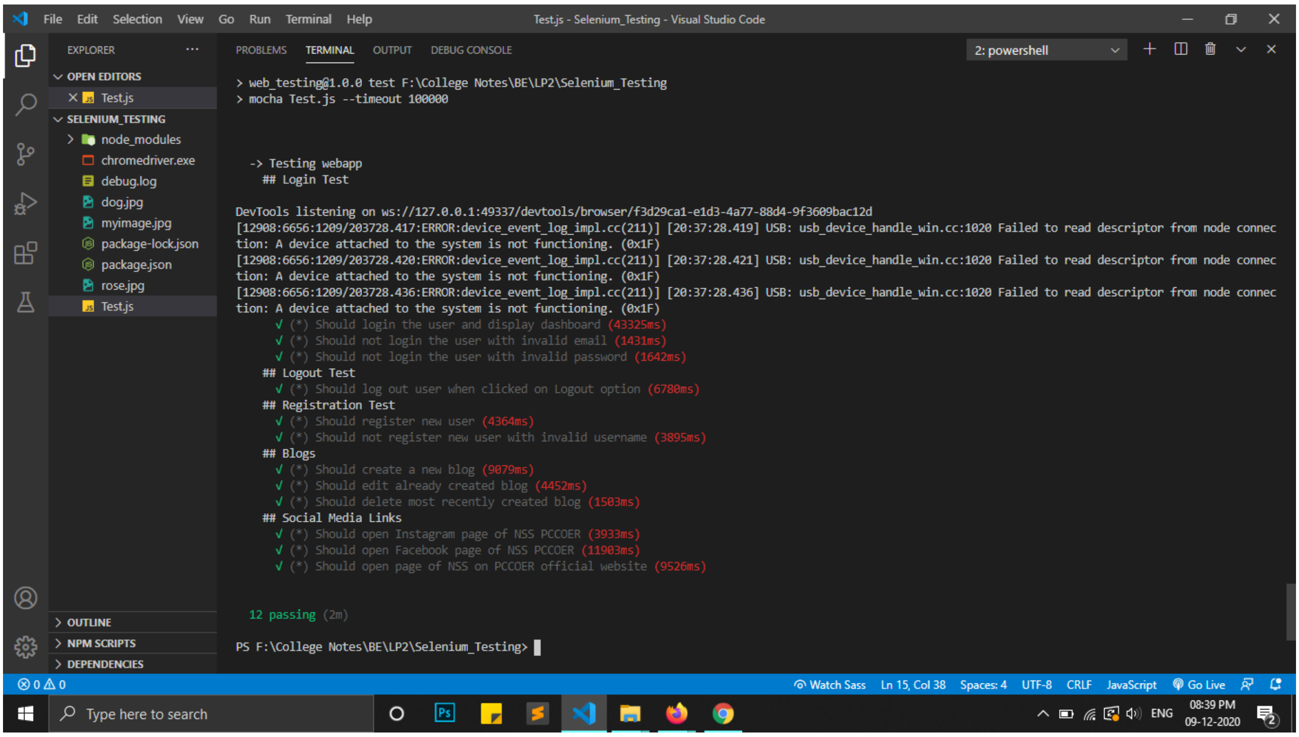
Submit

Figure 10.6: Add new blog/article


SPPU NSS PCCOER
Search results - sushantsaid1@
https://nss-pccoer-blog.herokuapp.com/blog/dashboard
90%
NATIONAL SERVICE SCHEME
SAVITRIBAI PHULE PUNE UNIVERSITY
PIMPRI CHINCHWAD EDUCATION TRUSTs
PIMPRI CHINCHWAD COLLEGE OF ENGINEERING AND RESEARCH
NSS PCCOER
Home
AddArticle
Logout
First
1
Last
Savitribai Phule Pune University
NATIONAL SERVICE SCHEME
Type here to search
08:16 PM
09-12-2020

s.no	title	author	views	tags	created	updated	edit/delete
1	Title of the blog	Akash		Some tags here	12/2/2020, 1:29:44 PM		edit delete

Figure 10.7: Edit/Delete option of a blog post



```
> web_testing@1.0.0 test F:\College Notes\BE\LP2\Selenium_Testing
> mocha Test.js --timeout 100000



  -> Testing webapp
    ## Login Test

DevTools listening on ws://127.0.0.1:49337/devtools/browser/f3d29ca1-e1d3-4a77-88d4-9f3609bac12d
[12908:6656:1209/203728.417:ERROR:device_event_log_impl.cc(211)] [20:37:28.419] USB: usb_device_handle_win.cc:1020 Failed to read descriptor from node connec
tion: A device attached to the system is not functioning. (0x1F)
[12908:6656:1209/203728.420:ERROR:device_event_log_impl.cc(211)] [20:37:28.421] USB: usb_device_handle_win.cc:1020 Failed to read descriptor from node connec
tion: A device attached to the system is not functioning. (0x1F)
[12908:6656:1209/203728.436:ERROR:device_event_log_impl.cc(211)] [20:37:28.436] USB: usb_device_handle_win.cc:1020 Failed to read descriptor from node connec
tion: A device attached to the system is not functioning. (0x1F)
      √ (*) Should login the user and display dashboard (43325ms)
      √ (*) Should not login the user with invalid email (1431ms)
      √ (*) Should not login the user with invalid password (1642ms)
    ## Logout Test
      √ (*) Should log out user when clicked on Logout option (6780ms)
    ## Registration Test
      √ (*) Should register new user (4364ms)
      √ (*) Should not register new user with invalid username (3895ms)
    ## Blogs
      √ (*) Should create a new blog (9079ms)
      √ (*) Should edit already created blog (4452ms)
      √ (*) Should delete most recently created blog (1503ms)
    ## Social Media Links
      √ (*) Should open Instagram page of NSS PCCOER (3933ms)
      √ (*) Should open Facebook page of NSS PCCOER (11903ms)
      √ (*) Should open page of NSS on PCCOER official website (9526ms)


  12 passing (2m)

PS F:\College Notes\BE\LP2\Selenium_Testing>
```

Figure 10.8: Testing Output

11 Conclusion

The NSS Blog web application is successfully implemented using Node.js and tested using Selenium. Selenium WebDriver is a powerful tool in Selenium Tool Suite, it supports various programming environments, various operating environments, and various popular browsers. Selenium is a very popular automation tool nowadays. It support many programming languages.

Thus, we have successfully tested our web application using Selenium WebDriver.

References

[1] https://www.selenium.dev/selenium/docs/api/javascript/index.html

[2] https://www.lambdatest.com/blog/mocha-javascript-tutorial-with-examples-for-selenium-testing/

[3] https://nehalist.io/selenium-tests-with-mocha-and-chai-in-javascript/