# PROFESSIONAL TRAINING REPORT

**at**

**Sathyabama Institute of Science and Technology (Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering

By

## SOMU.K.B

**REG. NO. 39110958**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,**

**CHENNAI – 600119, TAMILNADU**

**NOVEMBER 2021**

|  |  |  |
| --- | --- | --- |
|  | **SATHYABAMA**  **INSTITUTE OF SCIENCE AND TECHNOLOGY** (DEEMED TO BE UNIVERSITY) **Accredited with Grade “A” by NAAC**  (Established under Section 3 of UGC Act, 1956)  JEPPIAAR NAGAR, RAJIV GANDHI SALAI  CHENNAI– 600119  [**www.sathyabama.ac.in**](http://www.sathyabama.ac.in) |  |

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **SOMU.K.B**

**(Reg. No: 39110958)** who carried out the project entitled “**AUTOMATION OF ONLINE BUS RESERVATION**” under my supervision from June 2021 to November 2021.

## Internal Guide

## Mrs. Ramya G Franklin,M.E,

**Head of the Department**

**Dr.L. Lakshmanan,M.E,.Ph.D,**

**Dr.S. Vigneshwari,M.E,.Ph.D.**



## Submitted for Viva voce Examination held on

**Internal Examiner External Examiner**

**DECLARATION**

I, **SOMU.K. B** hereby declare that the project report entitled “**AUTOMATION OF ONLINE BUS RESERVATION**” done by me under the guidance of **Mrs. Ramya G Franklin,M.E,.Ph.D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

## DATE:

**PLACE: SIGNATURE OF THE CANDIDATE**

**ACKNOWLEDGEMENT**

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D**, **Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of **Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. J. Albert Mayan, M.E., Ph.D., for** his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

**TRAINING CERTIFICATE**

****

**ABSTRACT**

This project is about “AUTOMATION OF ONLINE BUS RESERVATION” is a web based system, This is the system for ticket booking in online remotely any type of location in Bangladesh and also helps to the bus owner to manage their business by using this system it's also helps to owner record their daily collected money from bus business and also helps to reduced paper to track any type of bus information such as ticket information passenger information, seat booking information any type of user can see the details of seat which are booked and which is available it's a automated process if any user book ticket in online before confirmation of ticket other user can book ticket no need to refresh page ,when user confirm ticket in other side in other hand they saw the seat colours is red its totally automated system not only see admin and operator but also seat details of bus passengers

The main purpose of our system if any user wants to go anywhere any location in our country, they want to use our system in online no need to go seat booking for bus counter they can select bus for his/her choose bus seat booking related any type of operation and any query they can using the system whole package are here if any user wants to go anywhere.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| CHAPTER No. | TITLE | PAGE No |
|  | DECLARATION | 3 |
|  | ACKNOWLEDGEMENT | 4 |
|  | TRAINING CERTIFICATE | 5 |
|  | ABSTRACT | 6 |
| 1 | INTRODUCTION | 8 |
|  | 1.1: - INTRODUCTION TO SELENIUM | 8 |
|  |  |  |
|  | 1.2: - SELENIUM IDE | 8 |
|  | 1.3: - SELENIUM COMMANDS | 9 |
|  | 1.4: - SELENIUM WEBDRIVER | 11 |
| 2 | AIM AND SCOPE OF THE PRESENT INVESTIGATION | 12 |
|  | 2.1: -AIM | 12 |
|  | 2.2: -PRESENT SCOPE AND INVESTIGATION | 12 |
| 3 | RESULTS AND DISCUSSION | 12 |
| 4 | SUMMARY AND CONCLUSIONS | 13 |
| 5 | REFERENCES | 14 |
| 6 | APPENDIX | 15 |
|  | A: - SCREENSHOTS | 15 |
|  | B: -OUTPUT | 18 |
|  | C: -SOURCECODE | 20 |

**1. INTRODUCTION**

**1.1**: INTRODUCTION TO SELENIUM

* Selenium is a robust set of took that supports rapid development of test automation for web-based applications.
* Selenium provides a rich set of testing functions specifically geared to the needs of testing of a web application.
* Selenium operations are highly flexible, allowing many options for locating UI elements and comparing expected test results against actual application behaviour.
* Features
* Supports Cross Browser Testing. The Selenium tests can be run on multiple browsers.
* Allows scripting in several languages like Java, C#, PHP and Python.
* Assertion statements provide an efficient way of comparing expected and actual results.
* Inbuilt reporting mechanism

**1.2**: SELENIUM IDE

The Selenium-IDE (Integrated Development Environment) is the tool you use to develop your Selenium test cases. It's an easy-to-use Firefox plug-in and is generally the most efficient way to develop test cases. It also contains a context menu that allows you to first select a UI element from the browser's currently displayed page and then select from a list of Selenium commands with parameters pre-defined according to the context of the selected UI element. This is not only a time-saver, but also an excellent way of learning Selenium script syntax Scripts may be automatically recorded and edited manually providing auto completion support and the ability to move commands around quickly. Scripts are recorded in Selenese, a special test scripting language for Selenium. Selenese provides commands for performing actions in a browser (click a link, select an option), and for retrieving data from the resulting pages

**1.3**: SELENIUM COMMANDS

To conclude our introduction of Selenium, we'll show you a few typical Selenium commands. These are probably the most commonly used commands for building tests.

* **open**

(Opens a page using a URL.)

* **click/clickAnd Wait**

(Performs a click operation, and optionally waits for a new page to load.)

* **verify Title/assert Title**

(Verifies an expected page title.)

* **verify TextPresent**

(Verifies expected text is somewhere on the page.)

* **verify Element Present**

(Verifies an expected UI element, as defined by its HTML tag, is present on the page.)

* **verify Text**

(Verifies expected text and is corresponding HTML tag are present on the page.)

* **verify Table**

(Verifies a table's expected contents.)

* **waitForPageToLoad**

**(**Pauses execution until an expected new page loads. Called automatically when click And Wait is used.)

* **waitForElementPresent**

**(**Pauses execution until an expected UI element, as defined by its HTML tag, is present on the page.)

**1.4:** SELENIUM WEBDRIVER



Selenium WebDriver is the successor to Selenium RC. Seknium WebDriver accepts commands (sent in Selenese, or via a Client API) and sends them to a browser. This is implemented through a browser-specific browser driver, which sends commands to a browser, and retrieves results. Most browser drivers actually launch and access a browser application (such as Firefox or Internet Explorer); there is also an HtmlUnit browser driver, which simulates a browser using HtmlUnit.

Unlike in Selenium 1, where the Selenium server was necessary to run tests, Selenium WebDriver does not need a special server to execute tests. Instead, the WebDriver directly starts a browser instance and controls it. However, Selenium Grid can be used with WebDriver to execute tests on remote systems.

In practice, this means that the Selenium 2.0 API has significantly fewer calk than does the Selenium 1.0 API. Where Selenium 10 attempted to provide a rich interface for many different browser operations, Selenium 2.0 aims to provide a basic set of building blocks from which developers can create their own Domain

Specific Language. One such DSL already exists: the Watir project in the Ruby language has a rich history of good design. Water-webdriver implements the Watir API as a wrapper for Selenium-Webdriver in Ruby. Watir-webdriver is created entirely automatically, based on the WebDriver specification and the HTML specification.

As of early 2012, Simon Stewart (inventor of WebDriver), who was then with Google and now with Facebook, and David Burns of Mozilla were negotiating with the W3C to make WebDriver an internet standard. In early 2013, the working draft was released. As such, Selenium Webdriver (Selenium 2.0) aims to be the reference implementation of the WebDriver standard in various programming languages. Currently Selenium-WebDriver is fully implemented and supported in Python, Ruby, Java, and C#.

The biggest change in Selenium recently has been the inclusion of the WebDriver API, driving browser natively as a user would use on a remote machine using the Selenium Server it marks a leap forward in terms of browser automation.

Selenium WebDriver fits in the same role as RC did, and has incorporated the original Ex 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.

Selenium 1.0+ WebDriver = Selenium 2.0

* WebDriver is designed in a simpler and more concise programming interface along with addressing some limitations in the Selgium-RC APL
* WebDriver is a compact Object-Oriented API when compared to Selenium1.0
* It drives the browser much more effectively and over comes the limitations of Selenium 1.x which affected our functional test coverage, like the file upload or download, pop-ups and dialogs barrier

**2.AIM AND PRESENT SCOPE AND INVESTEGATION**

**2.1:** AIM

TO CREATE AUTOMETION OF ONLINE BUS RESERVATION

**2.2:** PRESENT SCOPE AND INVESTEGATION

The objective of “Automation online bus reservation” is to easily track the information of all customers, departure bus station and seat information and bus destination

This system is helpful to business owners to maintain their business by using this web application. Owner every moment checks the departure information and seats booking information and also help to keeping record of every transaction by using this application not only business owner but also general user using this web page for the purpose of booking the tickets online general user no need to go bus counter to book tickets they can book ticket remotely anywhere in the world by fulfil this instruction.

**3.RESULTS AND DISCUSSION**

* First goes to online ticket booking website.
* Fills the from where box.
* Fills the to where box.
* Clicks on the calendar.
* Chooses a random future date from the calendar
* Clicks on search buses.
* First pop up will appear and closes the first popup.
* Second pop up will appear and closes the second popup.
* The list of buses will we shown which are present in from box and to box.
* Picks the first bus in the list and takes details of the bus.
* Details like boarding point and dropping point and the fare for a person in that bus and the bus facilities and amenities.
* Prints these details in eclipse console.

**4.SUMMARY AND CONCLUSIONS**

In most countries the use of road transportation is of high demand by citizens but some problems associated to this mean of transportation. Typical problems such as the manual booking of tickets from cashier at bus terminals by long Queue the scenarios and time consumption factor the tendency of misplacing bus tickets of customer and difficulty in retrieval off this record these problems can be solved by making life easier for the customers through the certain often online booking system at their convenience and comfortability this is main reason of developing such system.

The system is particularly of great importance because it reduces stress of queue to make tickets for booking and rental at the cashier counter to generate an accurate and detailed profile details and also management of booking seats effectively.

Any system which has been in use for a number of years gradually decays and becomes less effective because of the change in environment to which has to adopt for a time it is possible to overcome problems by amendment and minor modification 2 to acknowledge the need of fundamental changes computerization was proposed as a solution to the problems of being outdated with the fast present technologies in this project my aim to maximize my effort to computerize it accordingly that meet the entire passenger needs. I have successfully completed my project work on automation bus ticket reservation and the whole system has been done with simple data output obtained in according to the requirements.

**5.** **REFERENCES**

* WEBOGRAPHY

: -<http://google.com>

: -<http://docs.seleniumhq.org/>

: -<http://www.stackoverflow.com>

: -<https://poi.apache.org/apidocs/>

: -<http://browsermob-proxy-py.readthedocs.org/en/latest/>

* Selenium WebDriver in Eclipse IDE

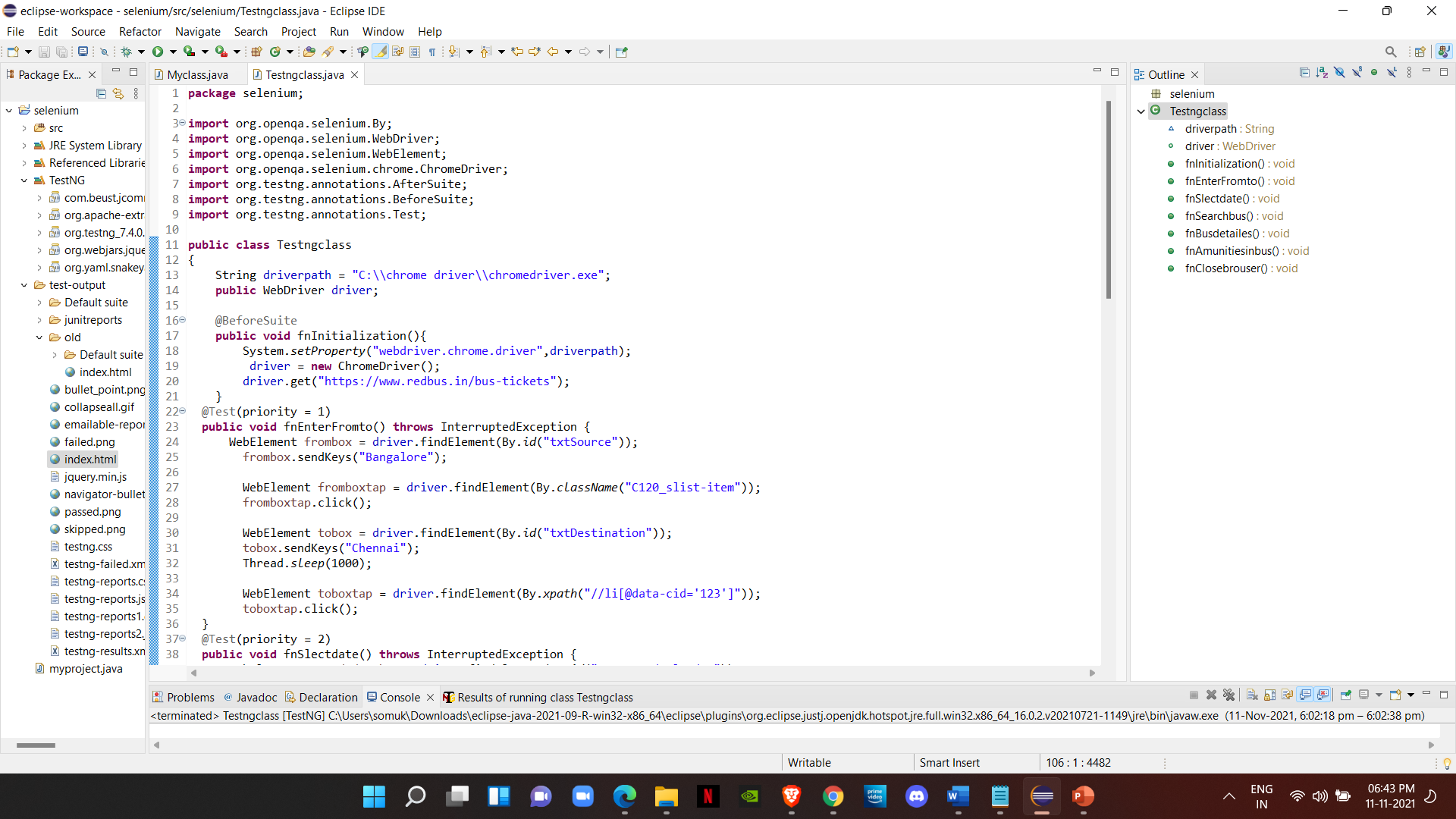
: - [*https://youtu.be/FVwSjBXb20o*](https://youtu.be/FVwSjBXb20o)

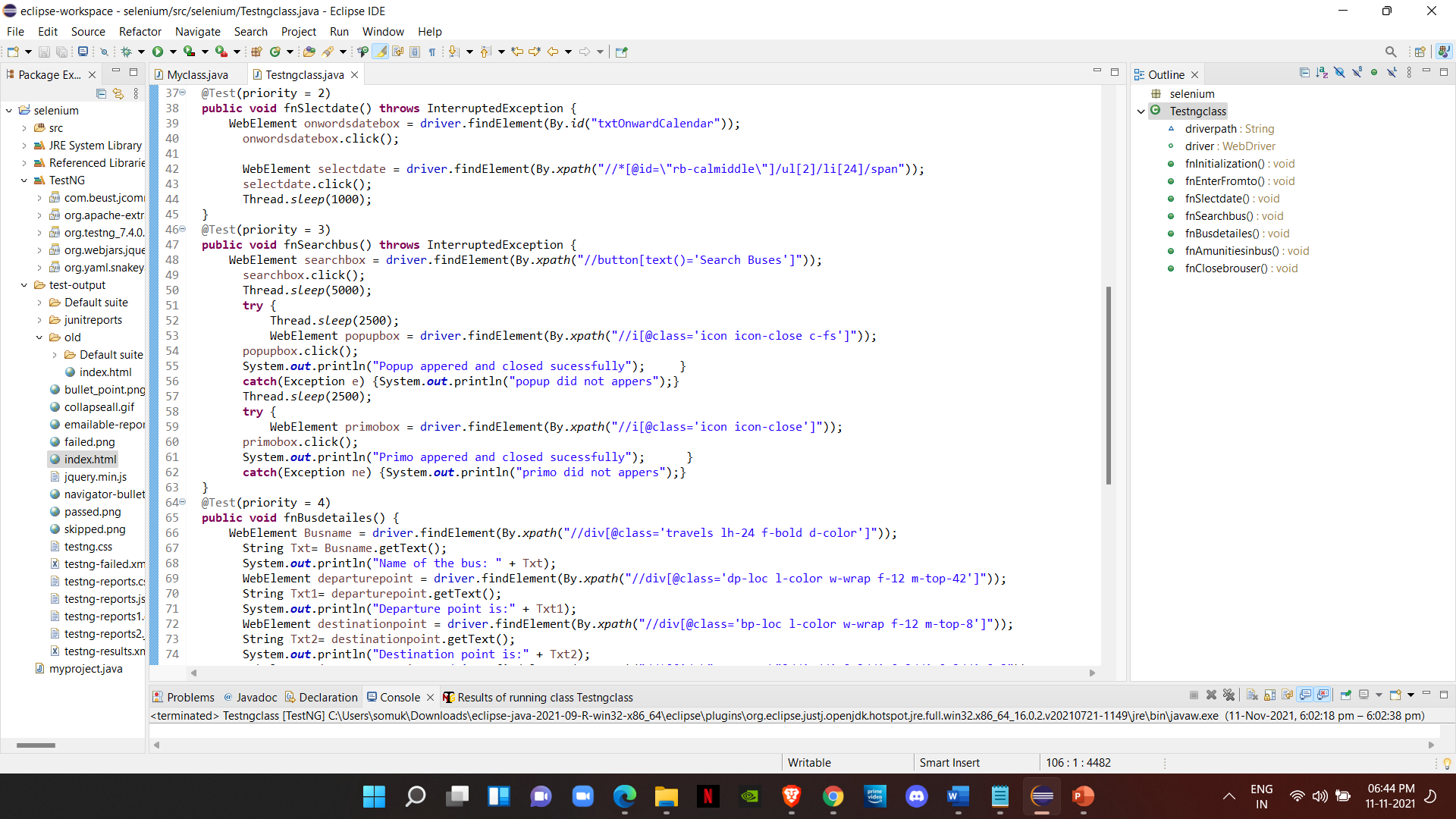
* Selenium testing tools

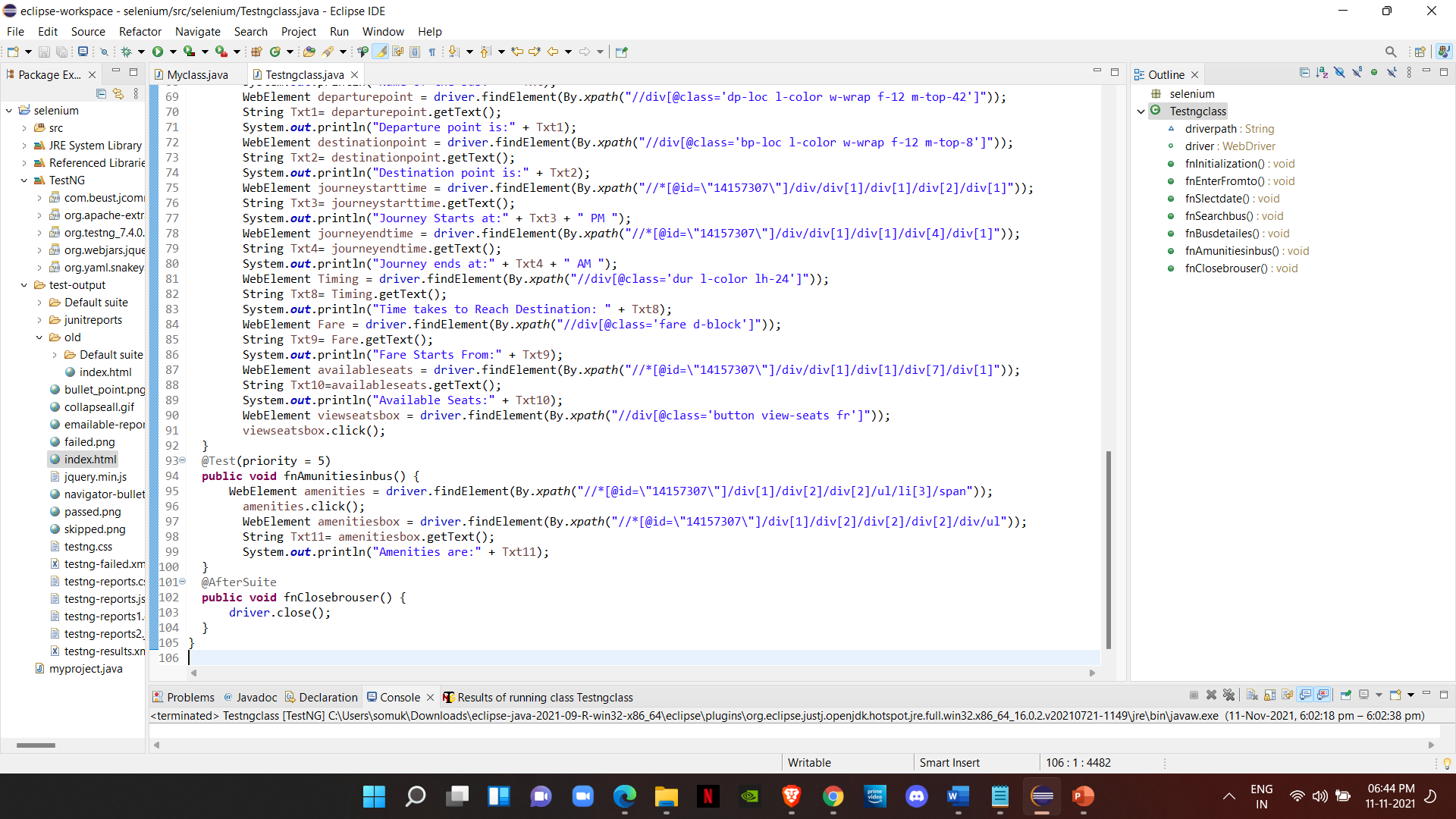
: - <https://www.geeksforgeeks.org/selenium-basics-components-features-uses-and-limitations/>

**6.** **APPENDIX**

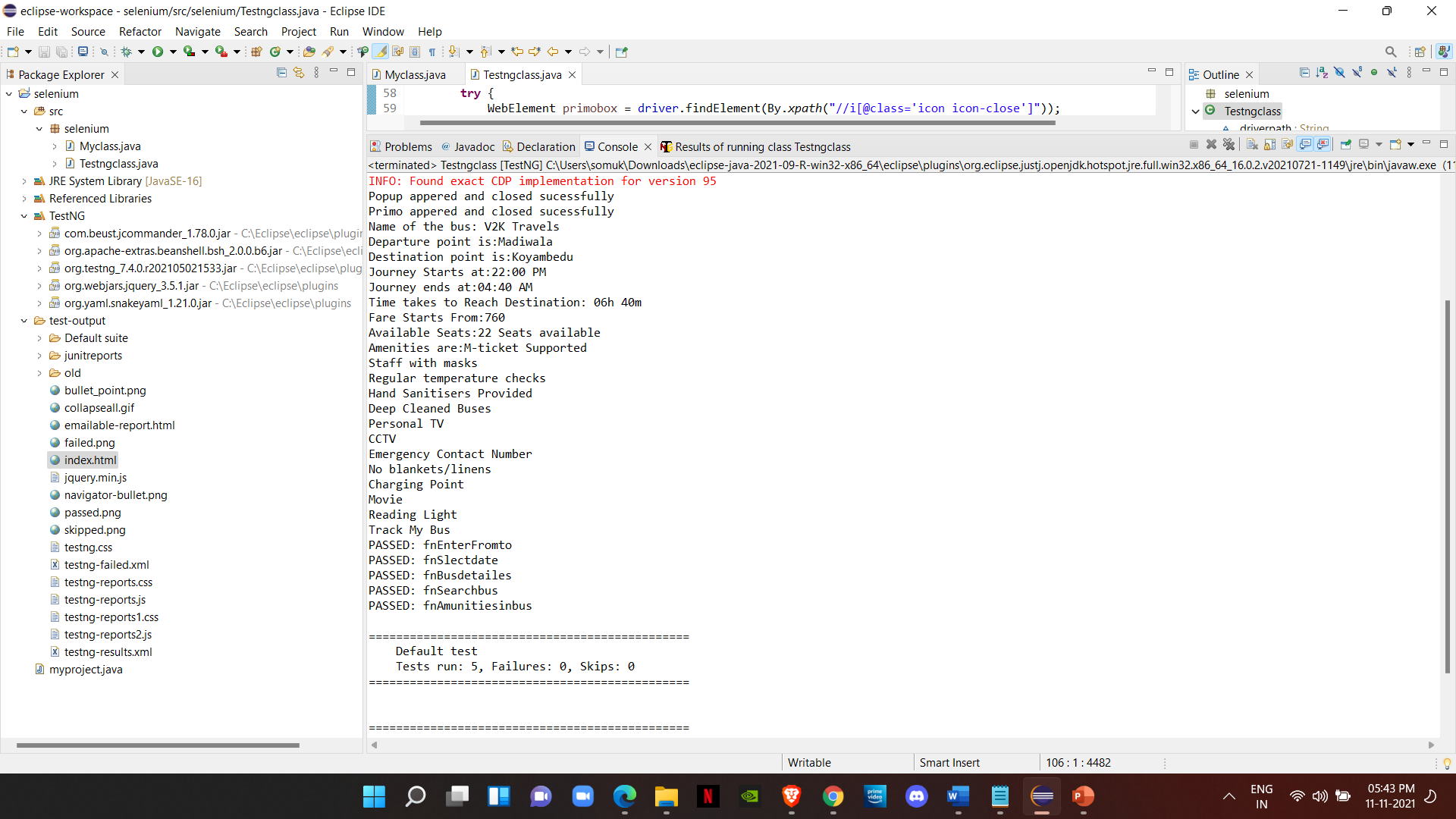
**A:** SCREENSHOTS

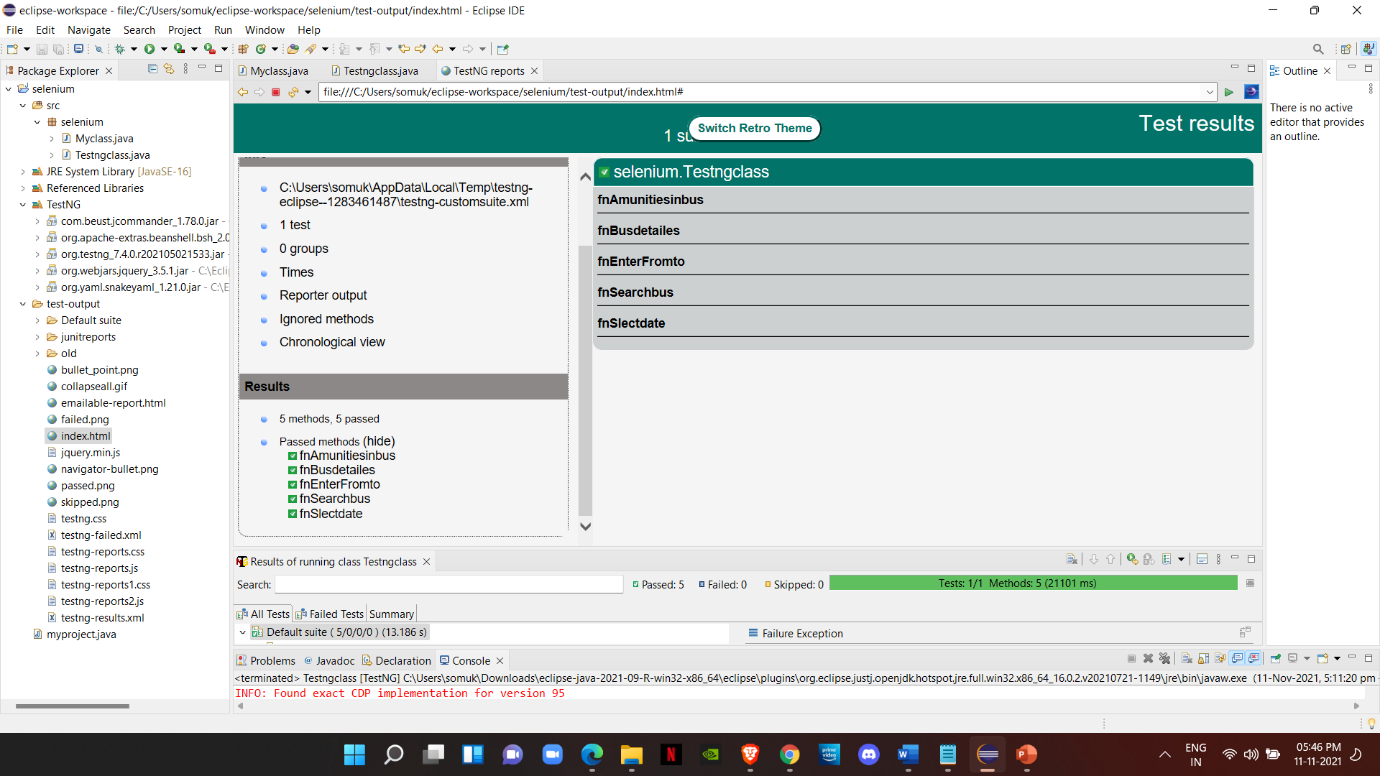


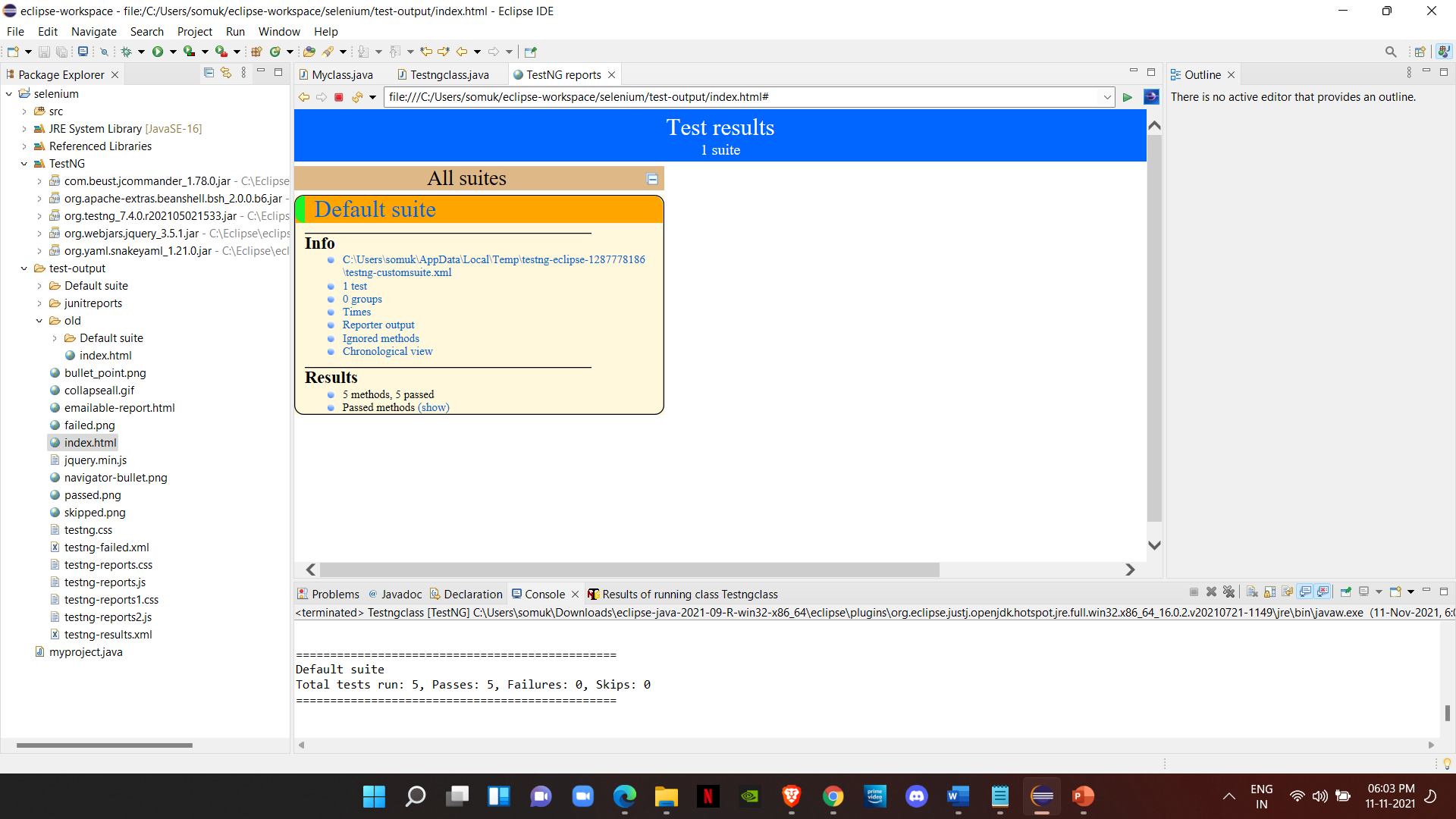




**B:** OUTPUT







**C:** SOURCE CODE

**package** selenium;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.testng.annotations.AfterSuite;

**import** org.testng.annotations.BeforeSuite;

**import** org.testng.annotations.Test;

**public** **class** Testngclass

{

String driverpath = "C:\\chrome driver\\chromedriver.exe";

**public** WebDriver driver;

@BeforeSuite

**public** **void** fnInitialization(){

System.*setProperty*("webdriver.chrome.driver",driverpath);

driver = **new** ChromeDriver();

driver.get("https://www.redbus.in/bus-tickets");

}

@Test(priority = 1)

**public** **void** fnEnterFromto() **throws** InterruptedException {

WebElement frombox = driver.findElement(By.*id*("txtSource"));

frombox.sendKeys("Bangalore");

WebElement fromboxtap = driver.findElement(By.*className*("C120\_slist-item"));

fromboxtap.click();

WebElement tobox = driver.findElement(By.*id*("txtDestination"));

tobox.sendKeys("Chennai");

Thread.*sleep*(1000);

WebElement toboxtap = driver.findElement(By.*xpath*("//li[@data-cid='123']"));

toboxtap.click();

}

@Test(priority = 2)

**public** **void** fnSlectdate() **throws** InterruptedException {

WebElement onwordsdatebox = driver.findElement(By.*id*("txtOnwardCalendar"));

onwordsdatebox.click();

WebElement selectdate = driver.findElement(By.*xpath*("//\*[@id=\"rb-calmiddle\"]/ul[2]/li[24]/span"));

selectdate.click();

Thread.*sleep*(1000);

}

@Test(priority = 3)

**public** **void** fnSearchbus() **throws** InterruptedException {

WebElement searchbox = driver.findElement(By.*xpath*("//button[text()='Search Buses']"));

searchbox.click();

Thread.*sleep*(5000);

**try** {

Thread.*sleep*(2500);

WebElement popupbox = driver.findElement(By.*xpath*("//i[@class='icon icon-close c-fs']"));

popupbox.click();

System.***out***.println("Popup appered and closed sucessfully"); }

**catch**(Exception e) {System.***out***.println("popup did not appers");}

Thread.*sleep*(2500);

**try** {

WebElement primobox = driver.findElement(By.*xpath*("//i[@class='icon icon-close']"));

primobox.click();

System.***out***.println("Primo appered and closed sucessfully"); }

**catch**(Exception ne) {System.***out***.println("primo did not appers");}

}

@Test(priority = 4)

**public** **void** fnBusdetailes() {

WebElement Busname = driver.findElement(By.*xpath*("//div[@class='travels lh-24 f-bold d-color']"));

String Txt= Busname.getText();

System.***out***.println("Name of the bus: " + Txt);

WebElement departurepoint = driver.findElement(By.*xpath*("//div[@class='dp-loc l-color w-wrap f-12 m-top-42']"));

String Txt1= departurepoint.getText();

System.***out***.println("Departure point is:" + Txt1);

WebElement destinationpoint = driver.findElement(By.*xpath*("//div[@class='bp-loc l-color w-wrap f-12 m-top-8']"));

String Txt2= destinationpoint.getText();

System.***out***.println("Destination point is:" + Txt2);

WebElement journeystarttime = driver.findElement(By.*xpath*("//\*[@id=\"14157307\"]/div/div[1]/div[1]/div[2]/div[1]"));

String Txt3= journeystarttime.getText();

System.***out***.println("Journey Starts at:" + Txt3 + " PM ");

WebElement journeyendtime = driver.findElement(By.*xpath*("//\*[@id=\"14157307\"]/div/div[1]/div[1]/div[4]/div[1]"));

String Txt4= journeyendtime.getText();

System.***out***.println("Journey ends at:" + Txt4 + " AM ");

WebElement Timing = driver.findElement(By.*xpath*("//div[@class='dur l-color lh-24']"));

String Txt8= Timing.getText();

System.***out***.println("Time takes to Reach Destination: " + Txt8);

WebElement Fare = driver.findElement(By.*xpath*("//div[@class='fare d-block']"));

String Txt9= Fare.getText();

System.***out***.println("Fare Starts From:" + Txt9);

WebElement availableseats = driver.findElement(By.*xpath*("//\*[@id=\"14157307\"]/div/div[1]/div[1]/div[7]/div[1]"));

String Txt10=availableseats.getText();

System.***out***.println("Available Seats:" + Txt10);

WebElement viewseatsbox = driver.findElement(By.*xpath*("//div[@class='button view-seats fr']"));

viewseatsbox.click();

}

@Test(priority = 5)

**public** **void** fnAmunitiesinbus() {

WebElement amenities = driver.findElement(By.*xpath*("//\*[@id=\"14157307\"]/div[1]/div[2]/div[2]/ul/li[3]/span"));

amenities.click();

WebElement amenitiesbox = driver.findElement(By.*xpath*("//\*[@id=\"14157307\"]/div[1]/div[2]/div[2]/div[2]/div/ul"));

String Txt11= amenitiesbox.getText();

System.***out***.println("Amenities are:" + Txt11);

}

@AfterSuite

**public** **void** fnClosebrouser() {

driver.close();

}

}