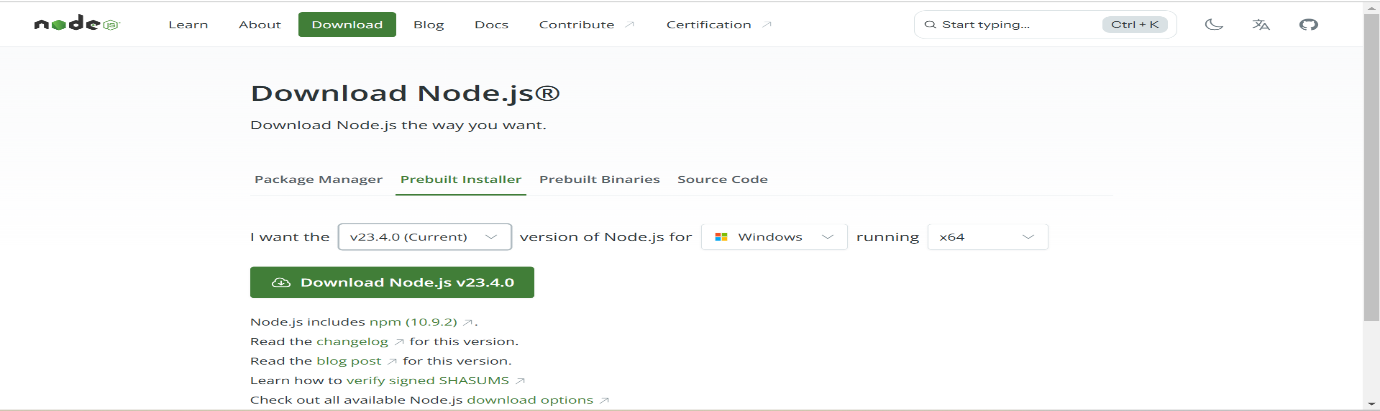
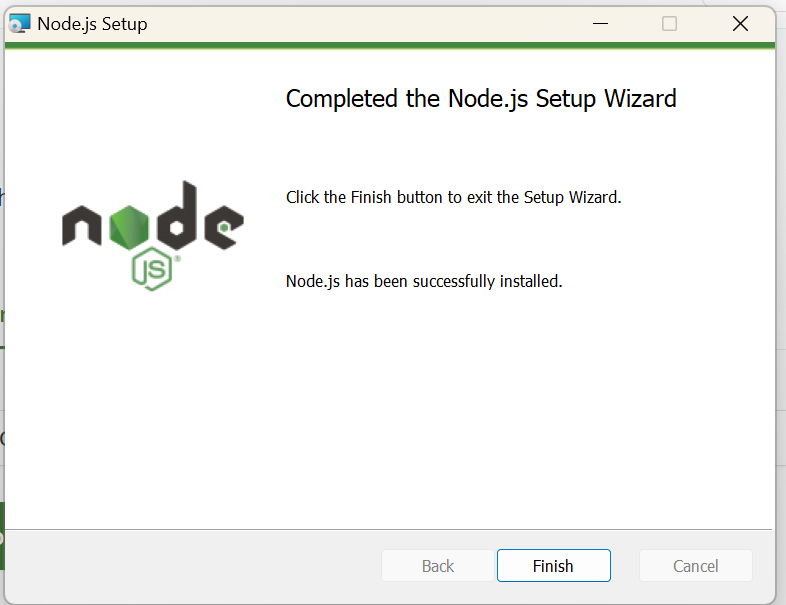
**EXPERIMENT-11:**

11.Write a simple progra in javascript and perform testing using selenium

* Download and install Node fro windows from chrome
* Double click on the downloaded file
* Click on next at every stage and enable the check boxes in the installation process
* Now create a folder named “sonar” and in that create another folder named “jmeter-demo”
* Open gitbash in this location and follow the commands

garid@Desk123 MINGW64 /d/sonar/jmeter-demo

$ mkdir googleTest

garid@Desk123 MINGW64 /d/sonar/jmeter-demo

$ cd googleTest/

garid@Desk123 MINGW64 /d/sonar/jmeter-demo/googleTest

$ node -v

v23.4.0

garid@Desk123 MINGW64 /d/sonar/jmeter-demo/googleTest

$ npm -version

10.9.2

garid@Desk123 MINGW64 /d/sonar/jmeter-demo/googleTest

$ npm init

This utility will walk you through creating a package.json file.

It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields

and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and

save it as a dependency in the package.json file.

Press ^C at any time to quit.

package name: (googletest)

version: (1.0.0)

description:

entry point: (index.js)

test command:

git repository:

keywords:

author:

license: (ISC)

About to write to D:\sonar\jmeter-demo\googleTest\package.json:

{

"name": "googletest",

"version": "1.0.0",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"author": "",

"license": "ISC",

"description": ""

}

Is this OK? (yes) yes

garid@Desk123 MINGW64 /d/sonar/jmeter-demo/googleTest

$ ls

package.json

garid@Desk123 MINGW64 /d/sonar/jmeter-demo/googleTest

$ npm install selenium-webdriver

added 17 packages, and audited 18 packages in 5s

1 package is looking for funding

run `npm fund` for details

found 0 vulnerabilities

garid@Desk123 MINGW64 /d/sonar/jmeter-demo/googleTest

$ cat package.json

{

"name": "googletest",

"version": "1.0.0",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"author": "",

"license": "ISC",

"description": "",

"dependencies": {

"selenium-webdriver": "^4.27.0"

}

}

garid@Desk123 MINGW64 /d/sonar/jmeter-demo/googleTest

$ npm install chromedriver

added 60 packages, and audited 78 packages in 11s

3 packages are looking for funding

run `npm fund` for details

found 0 vulnerabilities

garid@Desk123 MINGW64 /d/sonar/jmeter-demo/googleTest

$ ls

**node\_modules**/ package-lock.json package.json

garid@Desk123 MINGW64 /d/sonar/jmeter-demo/googleTest

$ cat package.json

{

"name": "googletest",

"version": "1.0.0",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"author": "",

"license": "ISC",

"description": "",

"dependencies": {

"chromedriver": "^131.0.3",

"selenium-webdriver": "^4.27.0"

}

}

garid@Desk123 MINGW64 /d/sonar/jmeter-demo/googleTest

$ npm install geckodriver

added 29 packages, and audited 107 packages in 5s

11 packages are looking for funding

run `npm fund` for details

found 0 vulnerabilities

garid@Desk123 MINGW64 /d/sonar/jmeter-demo/googleTest

$ cat package.json

{

"name": "googletest",

"version": "1.0.0",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"author": "",

"license": "ISC",

"description": "",

"dependencies": {

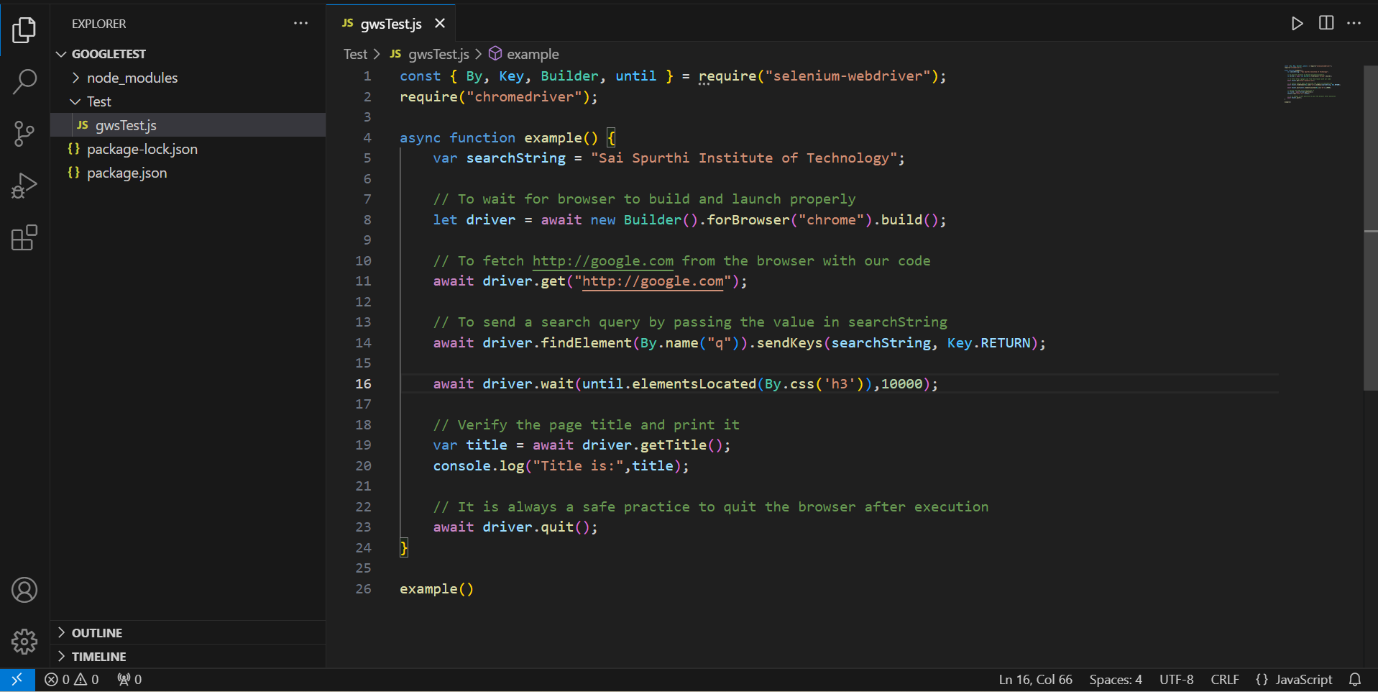
"chromedriver": "^131.0.3",

"geckodriver": "^5.0.0",

"selenium-webdriver": "^4.27.0"

}}

* Open Visual Studio Code
* Here open googleTest folder,in that create another folder named “Test”
* Create a file named gwsTest.js in the Test folder



* The javascript code is as follows

const { By, Key, Builder, until } = require("selenium-webdriver");

require("chromedriver");

async function example() {

    var searchString = "Sai Spurthi Institute of Technology";

    // To wait for browser to build and launch properly

    let driver = await new Builder().forBrowser("chrome").build();

    // To fetch http://google.com from the browser with our code

    await driver.get("http://google.com");

    // To send a search query by passing the value in searchString

    await driver.findElement(By.name("q")).sendKeys(searchString, Key.RETURN);

   // Wait for search results to load (Wait for the "first result" link to appear)

    await driver.wait(until.elementsLocated(By.css('h3')),10000);

    // Verify the page title and print it

    var title = await driver.getTitle();

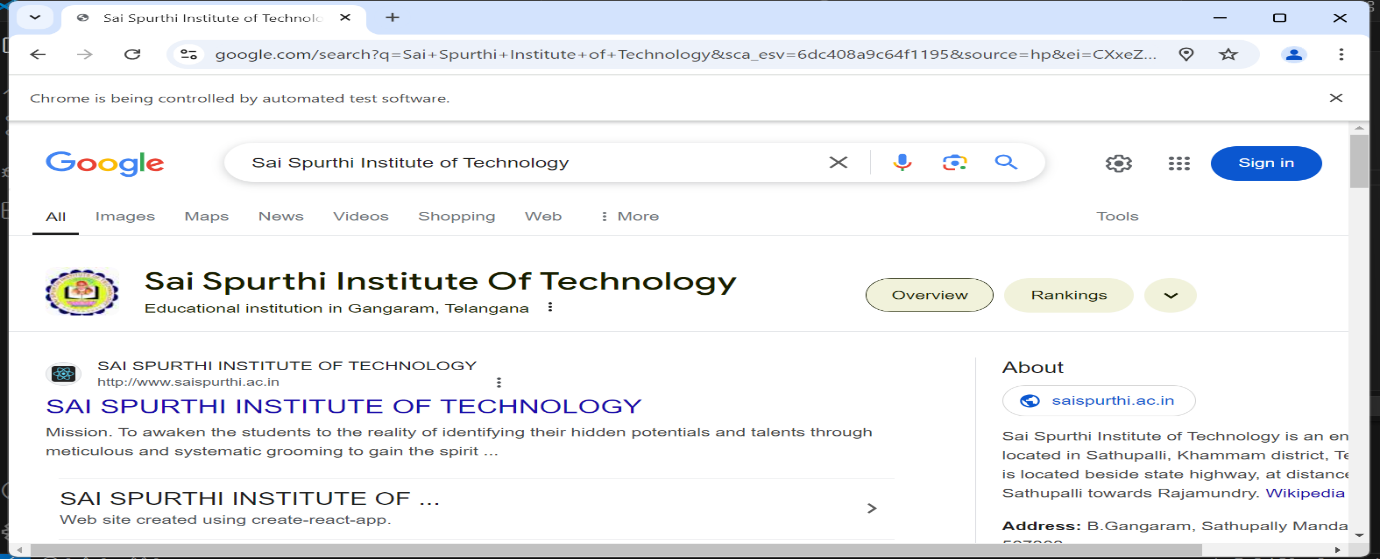
    console.log("Title is:",title);

    // It is always a safe practice to quit the browser after execution

    await driver.quit();

}

example()

**OUTPUT :**