St. Francis Institute of Technology, Mumbai-400 103

**Department Of Information Technology**

A.Y. 2024-2025

Class: TE-ITA/B, Semester: V

Subject: **DevOps Lab**

**Experiment – 8: To setup and run Selenium tests in Jenkins using Maven.**

1. **Aim:** To setup and run Selenium tests in Jenkins using Maven
2. **Objectives:** Aim of this experiment is that, the students will learn:

* Selenium and how to automate your test cases for testing web elements
* Introduction to X-Path, TestNG and integrate Selenium with Jenkins and Maven.

1. **Outcomes:** After study of this experiment, the students will learn following:

* Introduction to Selenium
* Installing Selenium
* Creating Test Cases in Selenium WebDriver
* Run Selenium Tests in Jenkins Using Maven

1. **Prerequisite:** Knowledge of Software Engineering concept of testing and test cases.
2. **Requirements:** Jenkins,JDK, Eclipse IDE, Firefox browser,Personal Computer, Windows operating system, Internet Connection, Microsoft Word.
3. **Pre-Experiment Exercise:**

**Brief Theory:** Refer shared material

1. **Laboratory Exercise**
   * + 1. **Procedure:**

**a. Answer the following:**

* Explain Selenium suite?
* What are the limitations of Selenium IDE?

**b**. **Execute following (Refer the shared material) and attach screenshots:**

* Create and run a test case on Chrome/Firefox browser with selenium IDE addon
* Create a Maven Project in Jenkins and run selenium tests using selenium Grid

1. **Post-Experiments Exercise**
2. **Extended Theory:**

Nil

1. **Questions:**

* What are Locators? Explain its types.
* What is the benefit of using Selenium Grid with Jenkins?

1. **Conclusion:**

* Write what was performed in the experiment.
* Write the significance of the topic studied in the experiment.

1. **References:**

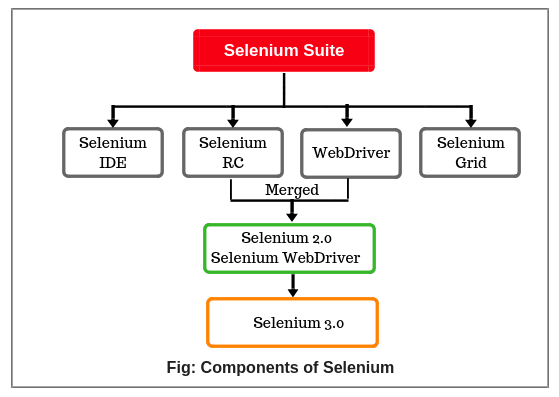
<https://jenkins.io/doc/>

<https://www.slideshare.net/abediaz/introduction-to-jenkins>

<https://q-automations.com/2019/09/26/selenium-grid-with-jenkins/>

A. Answer the following:

1. **Explain Selenium suite?  
   ANS:**Selenium is a popular open-source framework for automating web applications for testing purposes. **Selenium IDE**, a user-friendly browser extension for recording and playing back tests; and **Selenium Grid**, which allows for parallel execution of tests across multiple machines and browsers, enhancing testing efficiency. Together, these components provide a robust ecosystem for testing web applications, catering to both beginners and experienced testers.  
     
    It consists of several components:
2. **Selenium WebDriver:** This is the core component that allows you to write scripts in various programming languages (like Java, Python, C#, etc.) to control browsers programmatically. It provides a more flexible and powerful way to interact with web elements.
3. **Selenium IDE**: This is a browser extension (available for Firefox and Chrome) that allows for record-and-playback testing. It is user-friendly and ideal for beginners or for creating quick prototypes.
4. **Selenium Grid:** This component allows you to run tests on different machines and browsers simultaneously, facilitating parallel testing and reducing test execution time.
5. **Selenium RC (Remote Control):** An older component that has largely been replaced by WebDriver. It allows for more advanced testing, but it's less commonly used now.

****

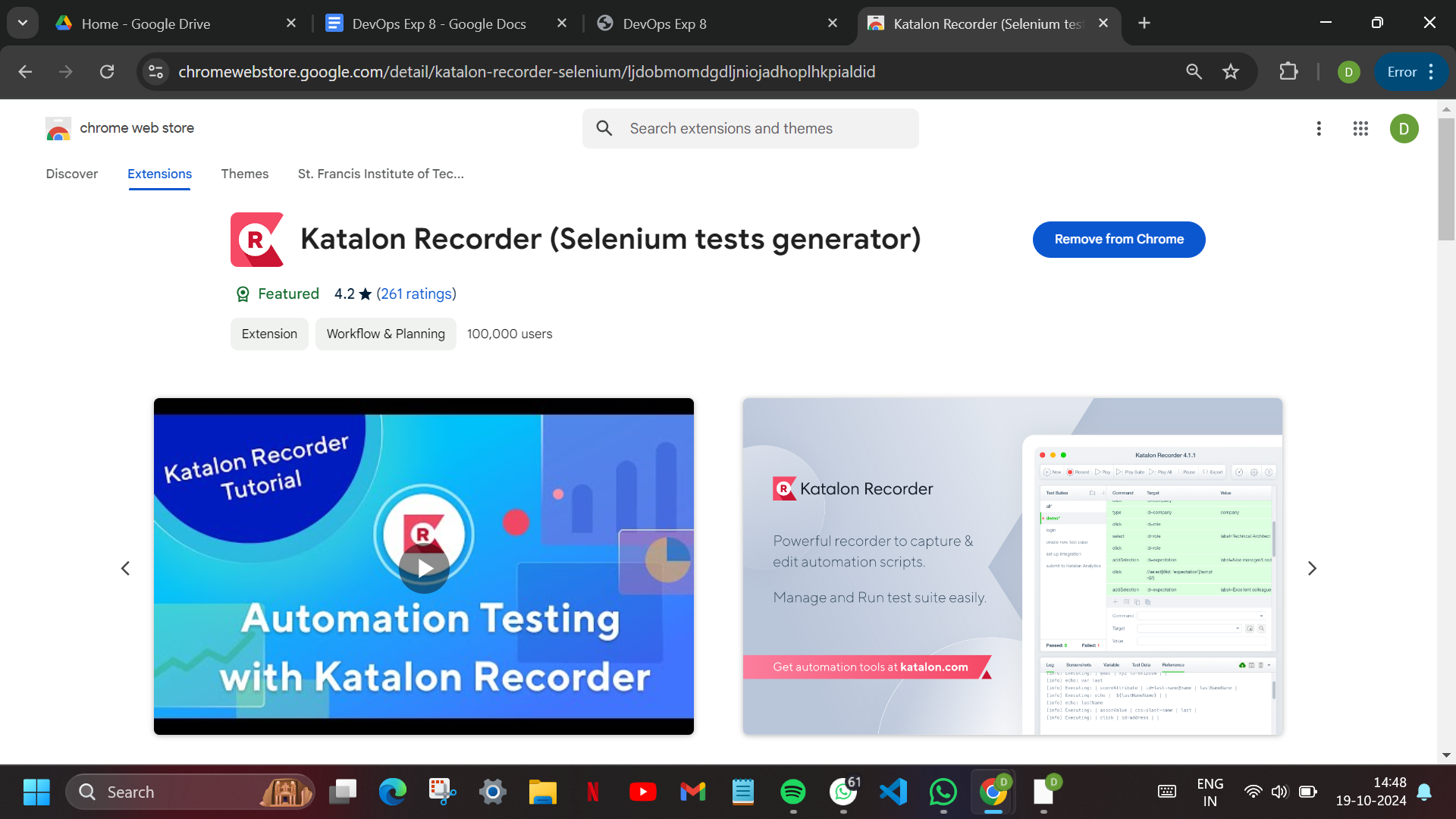
1. **What are the limitations of Selenium IDE?**

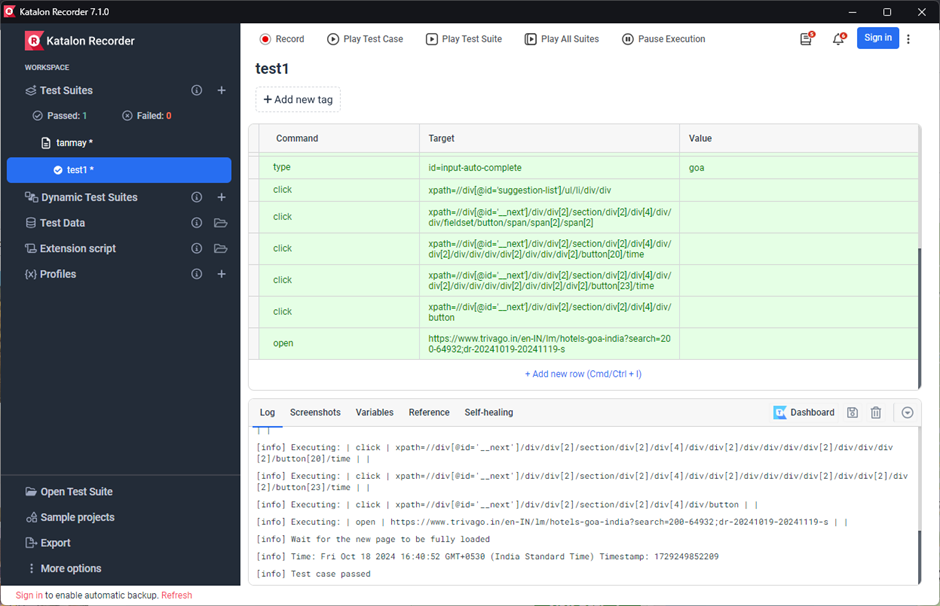
**ANS:**Limitations of Selenium IDE

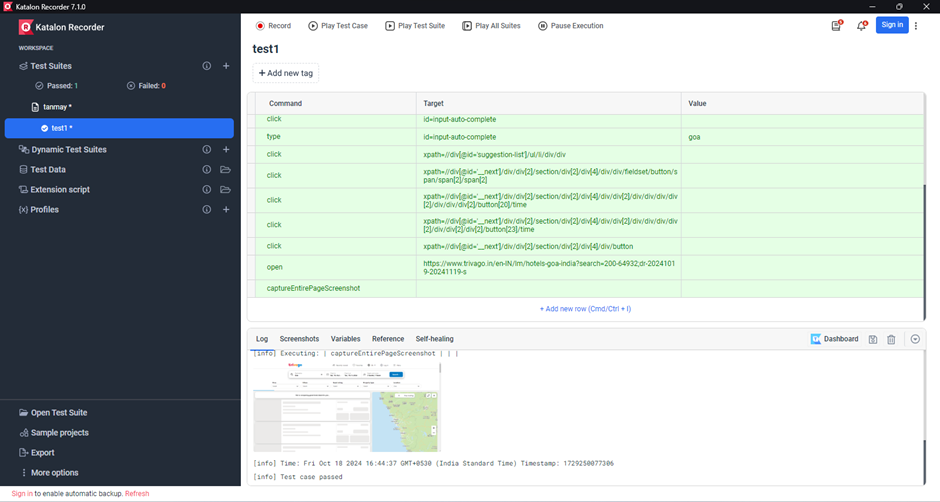
1. **Limited Functionality:** Selenium IDE is primarily for simple test automation and lacks advanced features needed for complex testing scenarios. It doesn’t support multi-threading or complex programming logic.
2. **Language Support**: Tests created in Selenium IDE are saved in its proprietary format (Selenese), making it difficult to integrate with other languages or frameworks.
3. **Dependency on the Browser:** Tests recorded in Selenium IDE are tied to the specific browser version and may not work if the browser is updated.
4. **No Support for Dynamic Content:** Selenium IDE struggles with dynamic web applications that heavily rely on AJAX or JavaScript changes.
5. **Limited Error Handling:** Error handling capabilities are basic, making it harder to manage failures gracefully.
6. **Lack of Extensibility:** Unlike WebDriver, Selenium IDE doesn't allow for adding custom commands or plugins to enhance functionality.
7. **Scalability Issues:** It’s not suitable for larger test suites that require maintenance, organization, or integration into CI/CD pipelines.

While Selenium IDE can be useful for quick and straightforward test automation, for more complex scenarios, the other components of the Selenium suite (like WebDriver) are often more suitable.

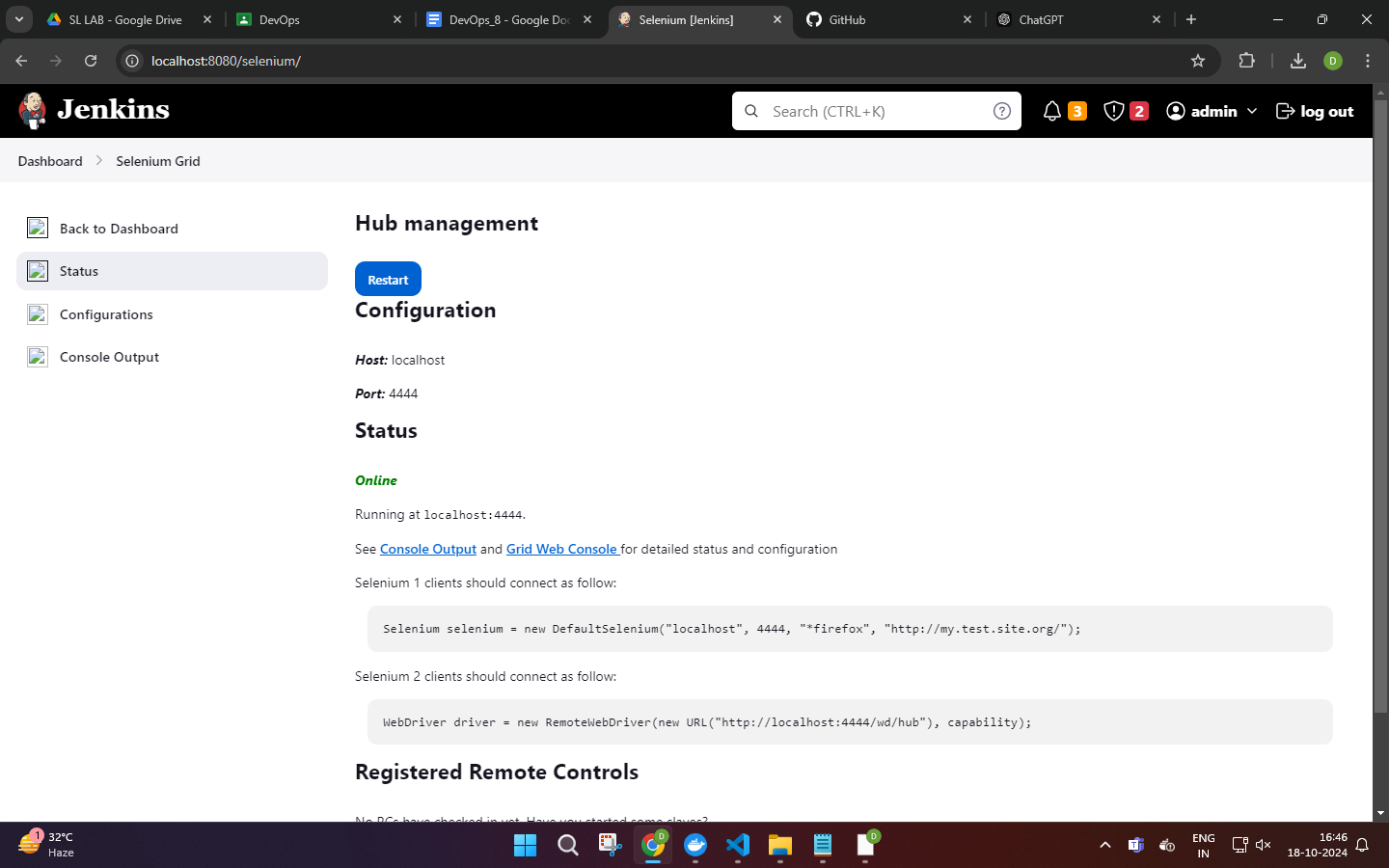
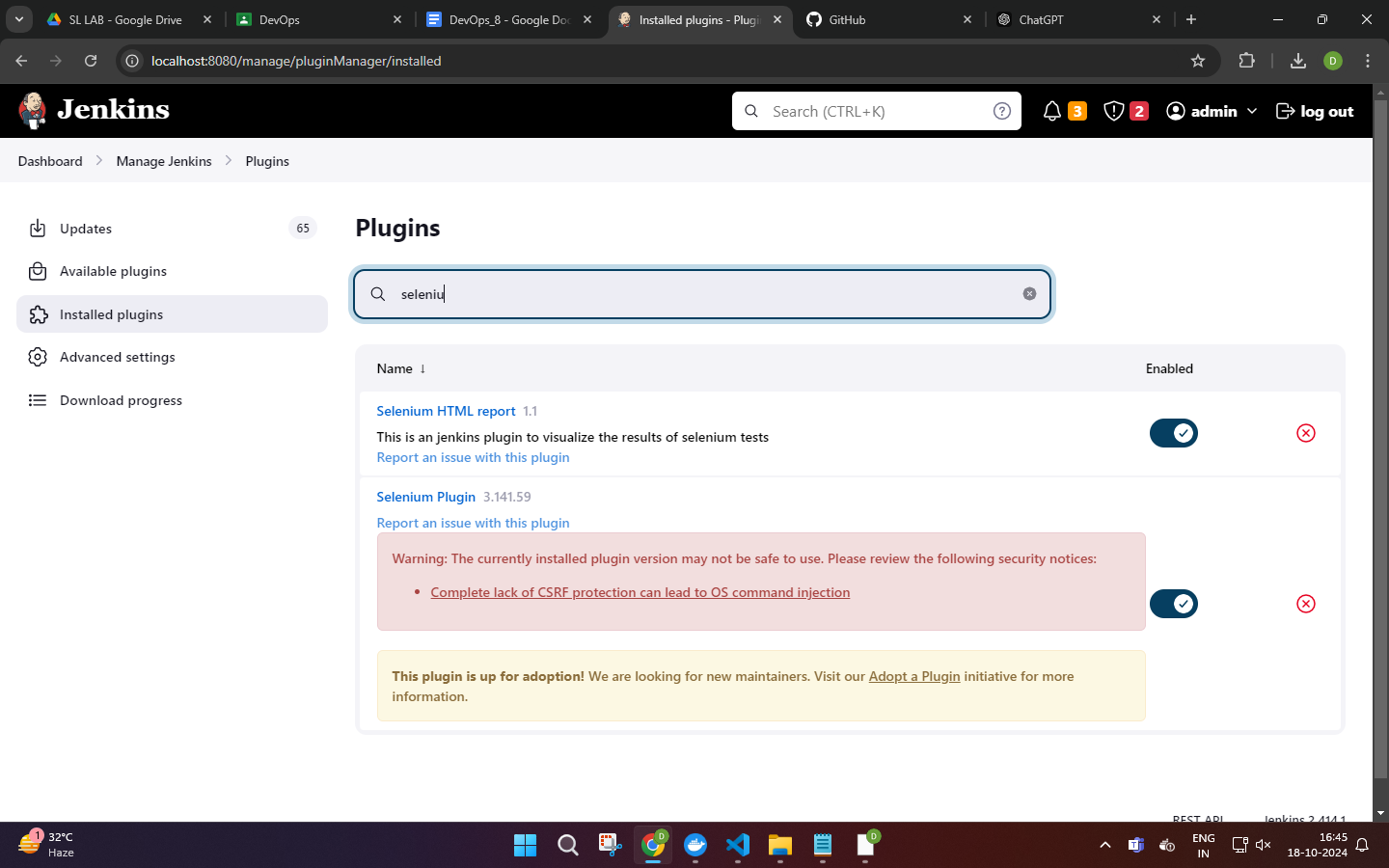
* **Create and run a test case on Chrome/Firefox browser with selenium IDE addon**

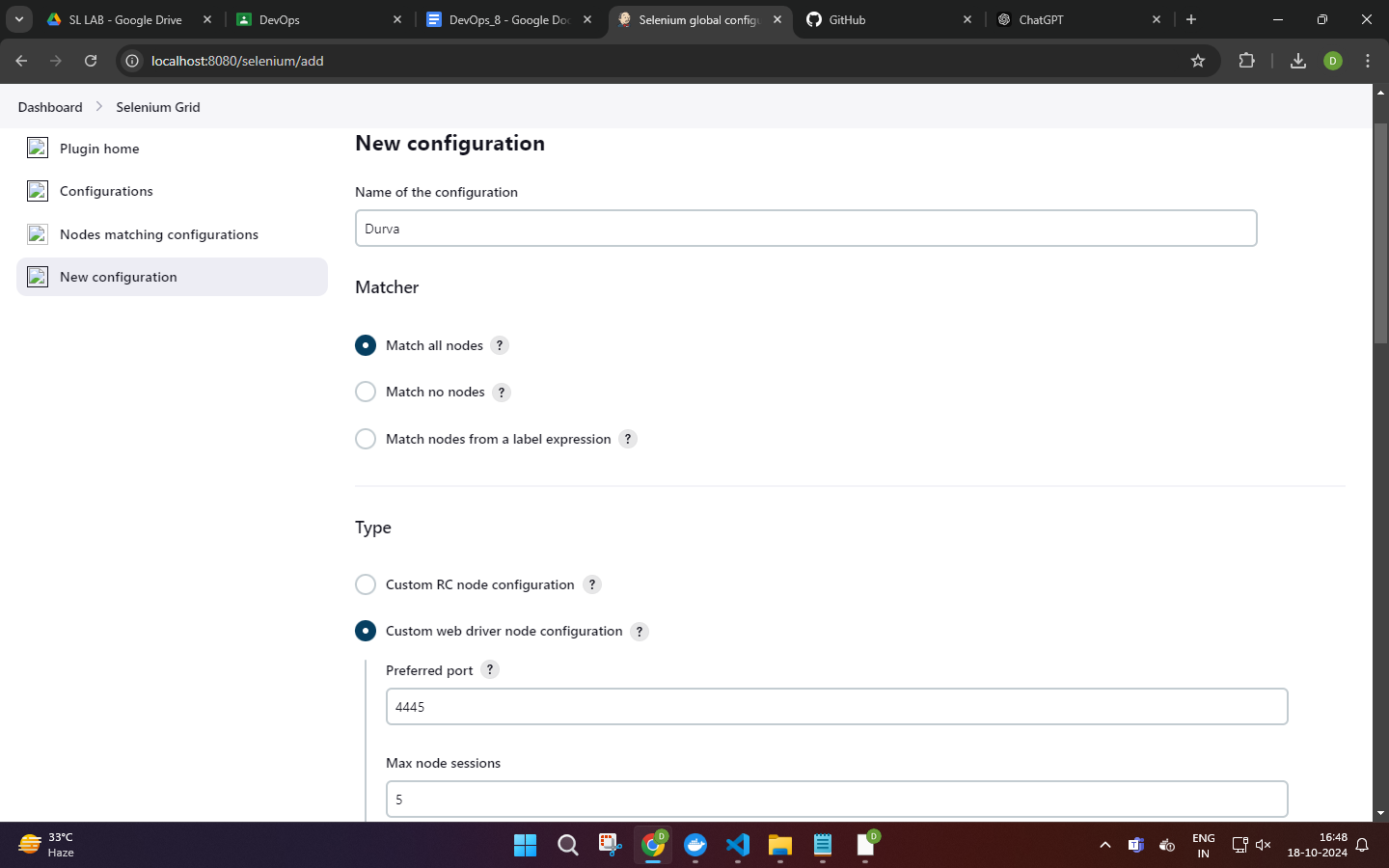


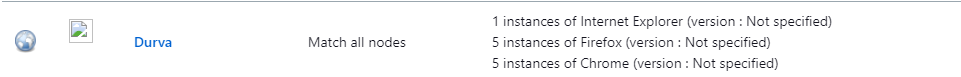


Getting a screenshot:

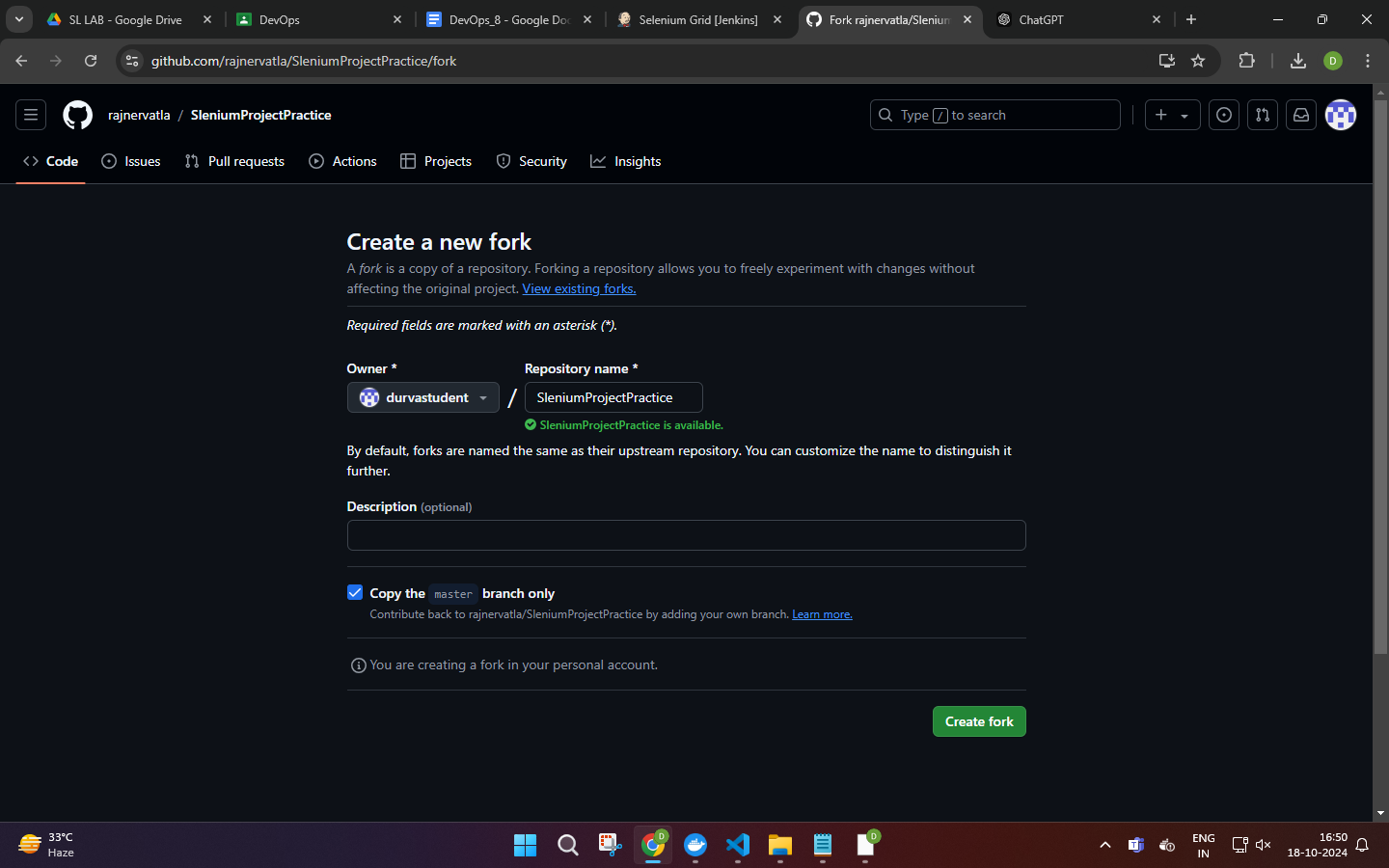
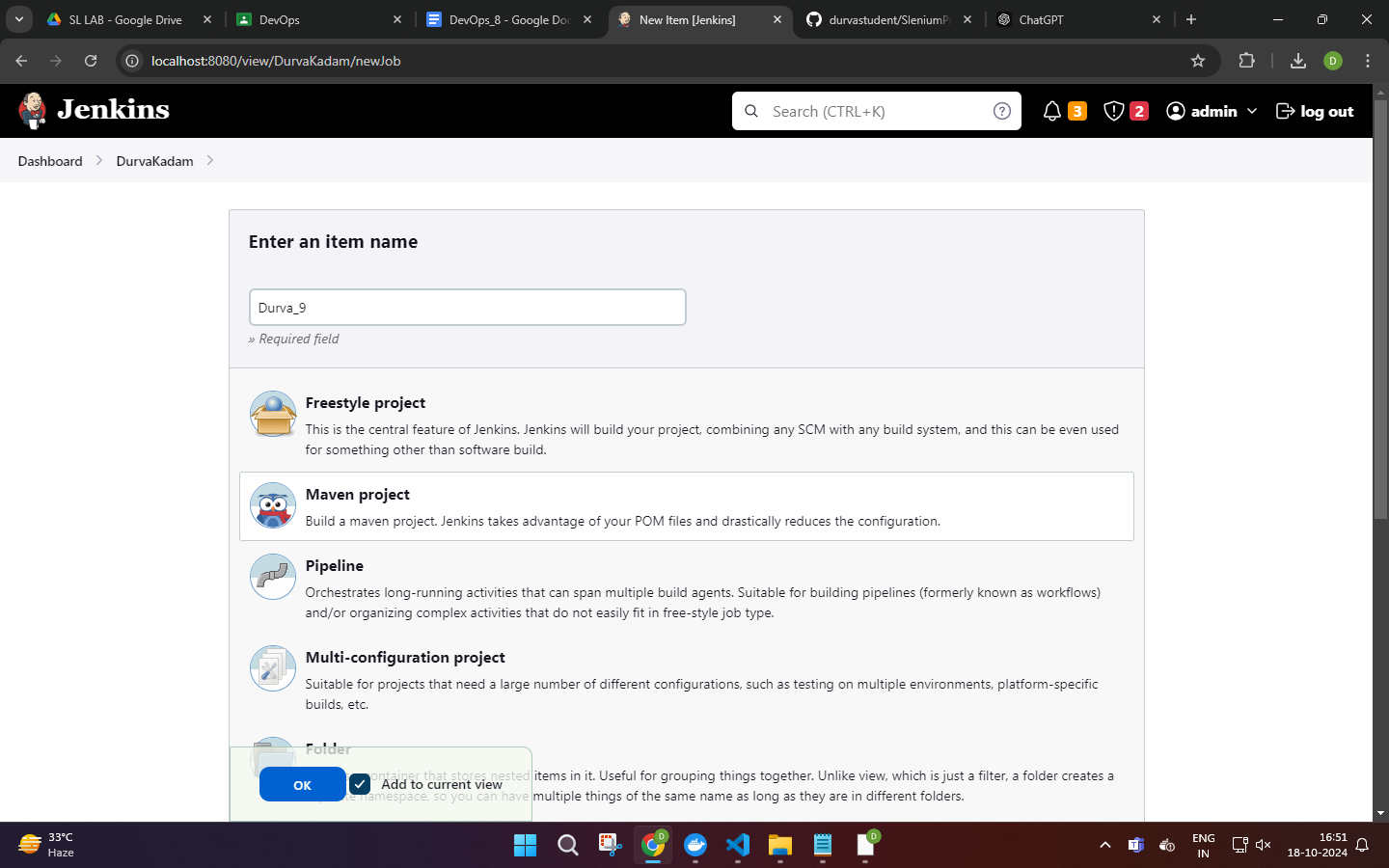
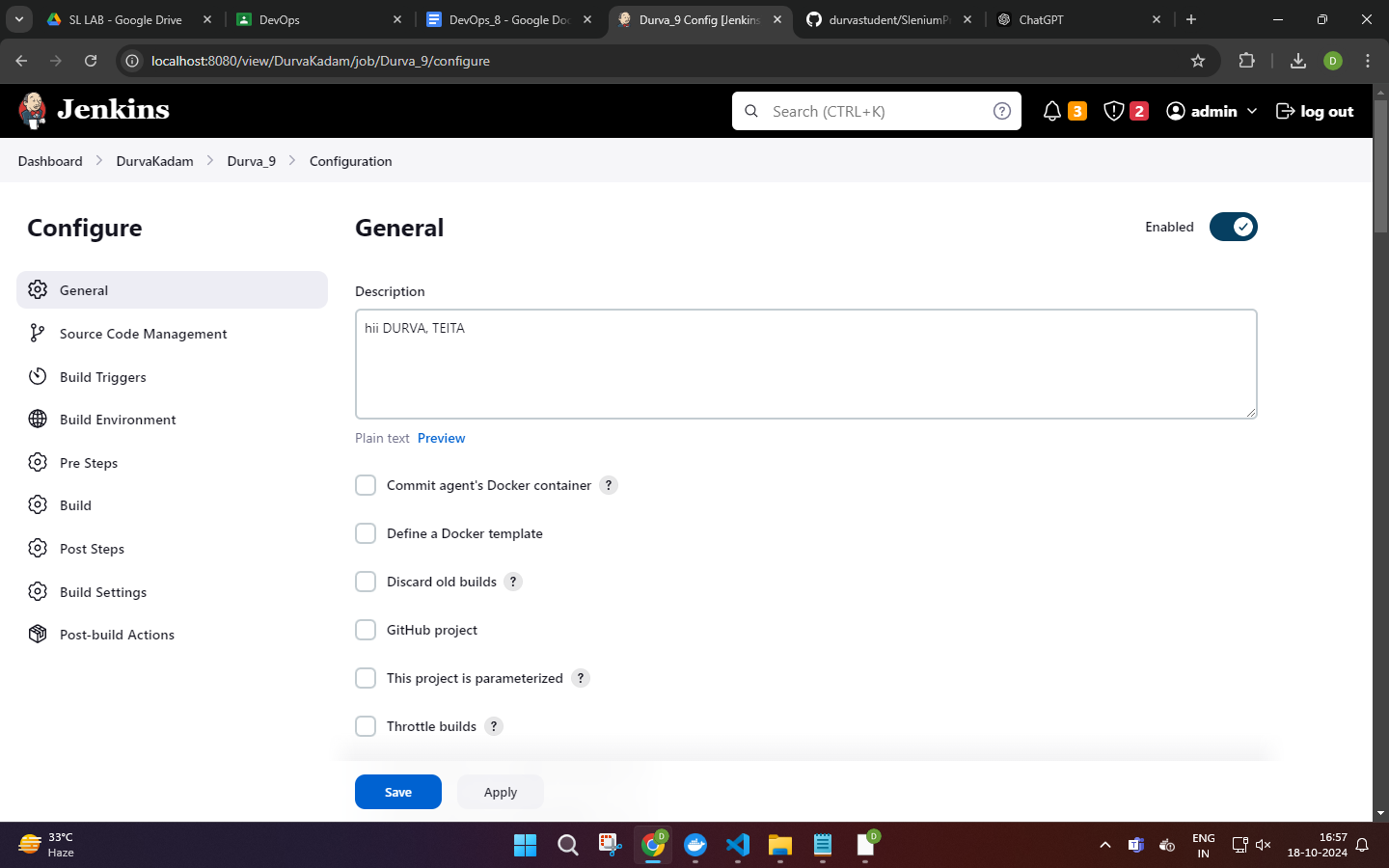
* Create a Maven Project in Jenkins and run selenium tests using selenium grid

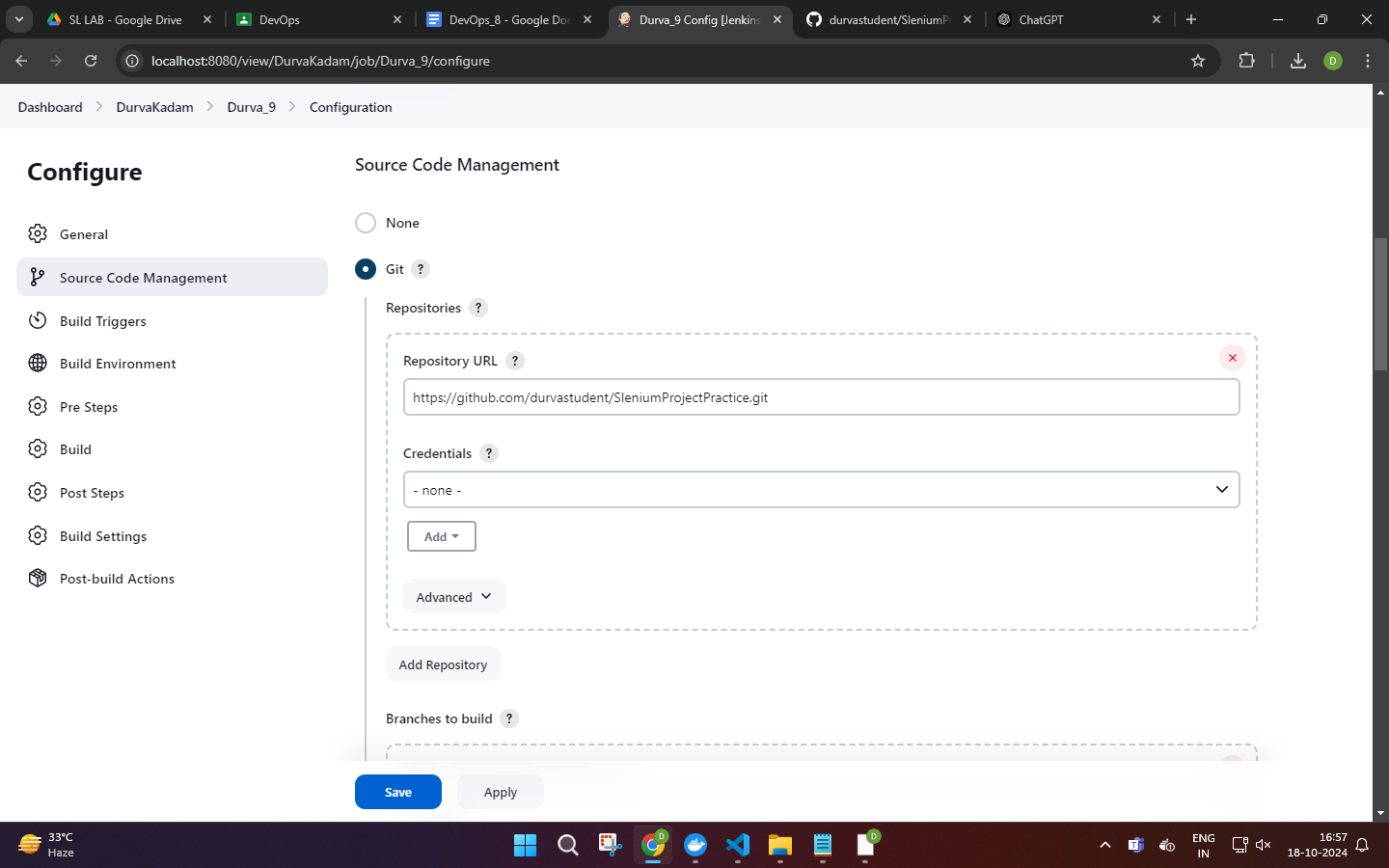


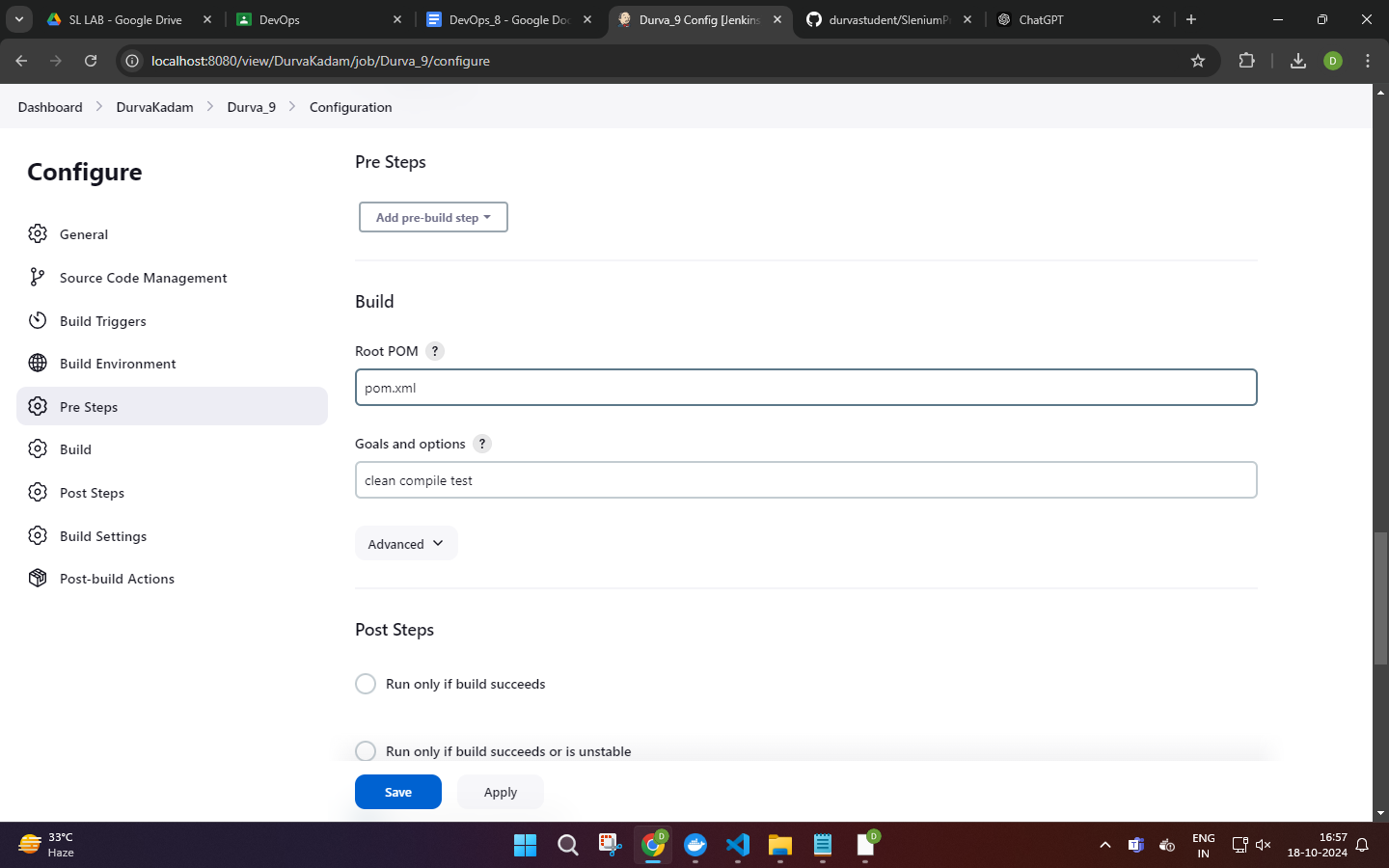
CREATING A NEW CONFIGURATION:

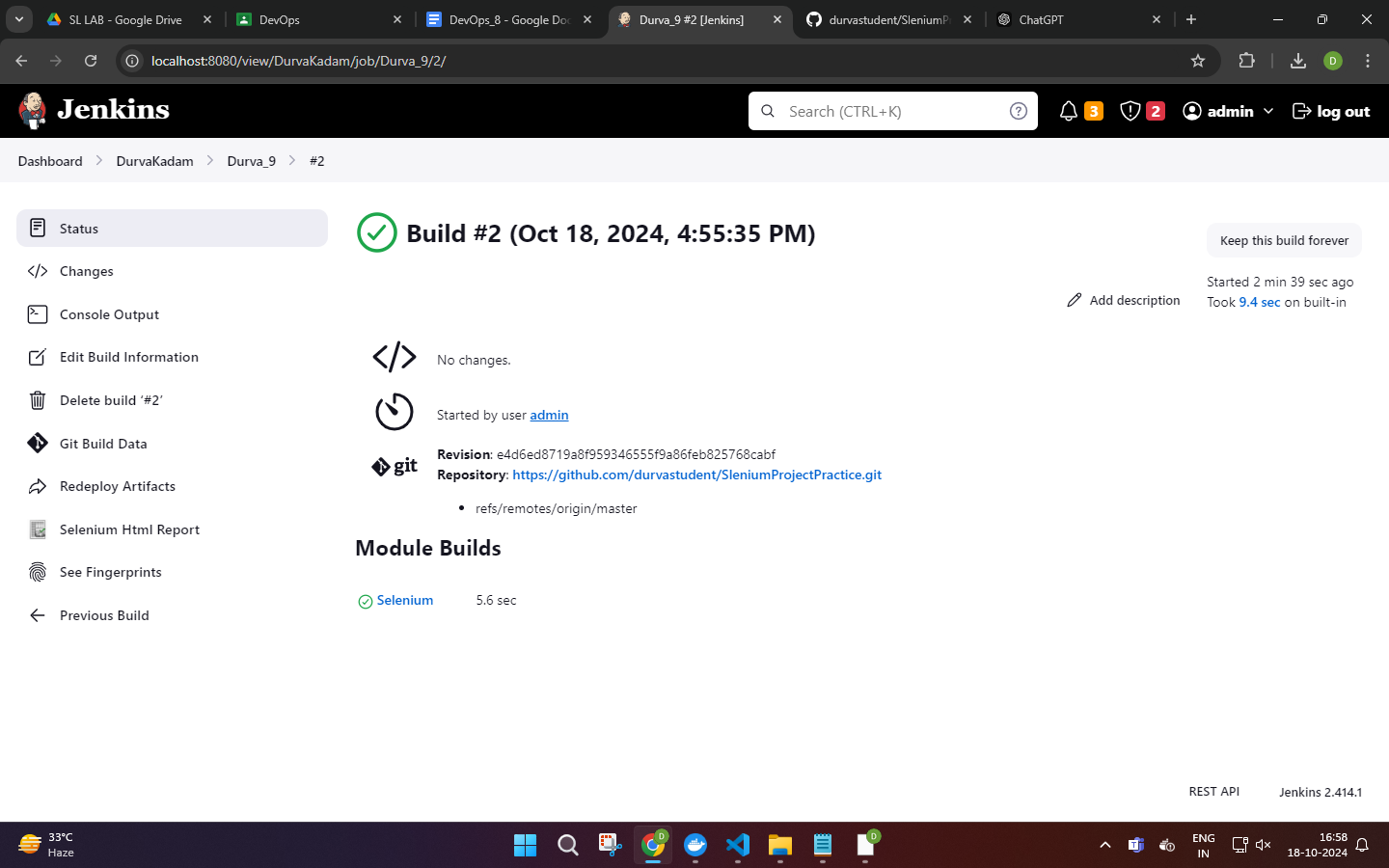


CREATING A FORK OF THE REPOSITORY:

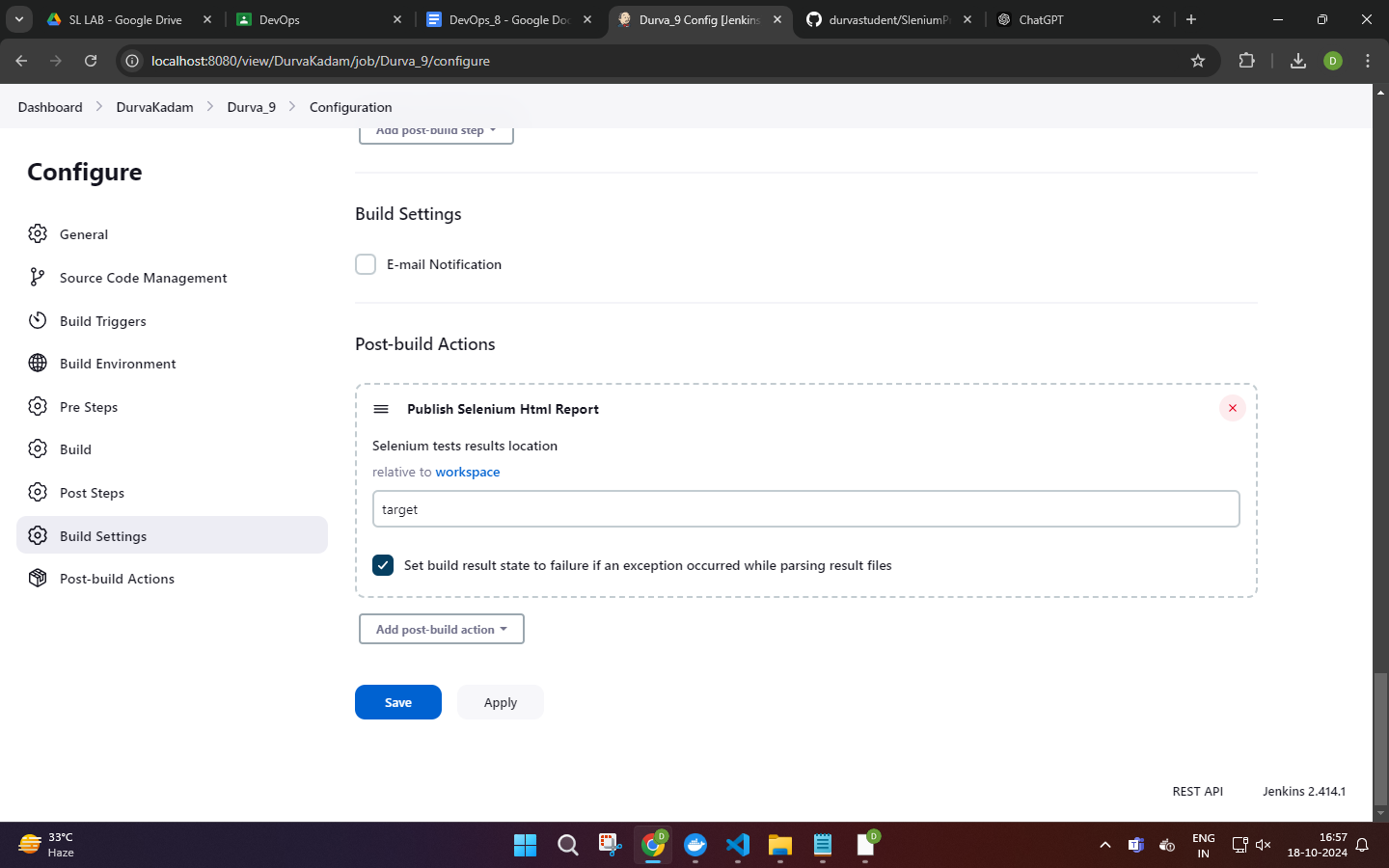
[https://github.com/rajnervatla/SleniumProjectPracticeCREATE A NEW ITEM IN VIEW:ADD GENERAL INFORMATION:](https://github.com/rajnervatla/SleniumProjectPractice*****)

[CONFIGURING THE THE SOURCE CODE:](https://github.com/rajnervatla/SleniumProjectPractice*****)

[POM MANAGEMENT:](https://github.com/rajnervatla/SleniumProjectPractice*****)

BUILD THE ITEM:

PUBLISH SELENIUM REPORT POST BUILD:



THE PUBLISHED REPORT:

