TEST AUTOMATION

EPAM Systems

A Training Report

Submitted in partial fulfilment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

FROM JAN 2023 TO APR 2023

SUBMITTED BY	**SUBMITTED TO**
Satyam Ranjan	**SAKSHI**
11917867	**Assistant Professor**

Student Declaration

To whom so ever it may concern

I, Satyam Ranjan, 11917867, hereby declare that the work done by me on "**TEST AUTOMATION**" from **JAN 2023** to **APR 2023**, under the supervision of **SAKSHI** and Lovely Professional University, Phagwara, Punjab, is a record of original work for the partial fulfilment of the requirements for the award of the degree Computer Science and Engineering.

Name: Satyam Ranjan
Registration number: 11917867

Satyam Ranjan
Signature of Student
Dated: 01/05/2023

Table of Contents

CHAPTER 1
ABOUT THE COMPANY

1.1 Company's Vision and Mission

Our teams of technologists, strategists and designers transform our customers' business through a combination of engineering expertise, design thinking and business consulting. The vision statement for EPAM Systems Inc. is its strategic plan for the future – it defines what and where EPAM Systems Inc. Company wants to be in the future. The vision statement for EPAM Systems Inc. is a document identifying the goals of EPAM Systems Inc. to facilitate its strategic, managerial, as well as general decision-making processes.

The vision statement of EPAM Systems Inc. is brief and to the point. This means that the company has not used long dialects and dialogues to delivers its opinion ad stance to the public and relevant stakeholders. The vision statement should be brief and comprehensive – it should communicate the essence of the business, and its future plans to help the stakeholders understand its business philosophy and business strategy.

The vision statement of EPAM Systems Inc. should be brief but should be holistic in nature. This means that the visions statement should be complete in its description and information of what the company desires, and how it plans to achieve its long-term goals strategically. The vision statement should be a comprehensive statement identifying the company's core strengths, which would enable it to achieve its futuristic goals.

The mission statement for EPAM Systems Inc. is a public document that details the values and strategic aims of EPAM Systems Inc. The mission statement of EPAM Systems Inc also identifies the purpose of the organization existence, highlighting the services and the products it offers. Further, the mission statement also identifies the organization's operational goals for EPAM Systems Inc. the processes the company uses to achieve those, the target customer groups, and the region where the company operates.

The mission statement of EPAM Systems Inc. focuses on addressing issues of customer satisfaction. The mission statement of EPAM Systems Inc. has identified its target customer groups, and also identified their needs and demands. The mission statement reflects on how its products and services work towards increasing customer satisfaction for its target customers.

The mission statement of EPAM Systems Inc. is based on its integral strengths and competencies. This is important for EPAM Systems Inc. as the mission statement will highlight the different systems and processes as well as strategic tactics that the company uses to achieve its organizational and strategic goals. The achievement of the goals will depend on how well EPAM Systems Inc. makes use of its core competencies.

The mission statement for EPAM Systems Inc. is also realistic and clear. This means that EPAM Systems Inc. has used simple, string, and easily understood words and phrases in the drafting of its mission statement. Clarity is important so that the mission statement is understood by all relevant stakeholders of EPAM Systems Inc. Company. EPAM Systems Inc.'s mission statement is also realistic, which makes it able to achieve various set goals and targets.

The mission statement of EPAM Systems Inc. is motivational in that it works towards inspiring the employees and the workforce towards giving their optimal best performance towards the goal achievement of EPAM Systems Inc. The mission statement of EPAM Systems Inc. is also inspirational in that it develops the need for growth and progress in individuals – for the betterment of not only the company but also for their own selves.

1.2 Origin and Growth of Company

In 1993, Arkadiy Dobkin and Leo Lozner founded EPAM, a global software engineering services company, in New Jersey, USA and Minsk, Belarus.

In 2002, EPAM was ranked publicly for the first time as a fast-growing company by Deloitte & Touche.

In 2012, EPAM was listed on New York Stock Exchange under moniker EPAM, becoming the first company from Belarus on NYSE.

Timeline

In 2004, EPAM acquired Fathom Technology, a software development services company based in Budapest, Hungary, expanding its offshore services beyond North America. A couple years later, EPAM secured an equity investment from Siguler Guff to fund its competitive growth plans.

In 2006, EPAM acquired VDI, a software development services company with delivery centers in Russia, which expanded the company's presence in the CIS region. That year, EPAM CEO Arkadiy Dobkin was named one of the Top 25 Most Influential Consultants of the Year by Consulting Magazine.

In late 2012, EPAM made two acquisitions – Thoughtcorp, which expanded its service offerings in Agile, business intelligence and mobile, and Empathy Lab, which established a digital engagement practice focusing on customer experience, design and eCommerce.
EPAM made two acquisitions in 2018 to expand its service offerings: Continuum (now EPAM Continuum) and TH_NK to add consulting capabilities and develop its digital and service design practices. Also that year, EPAM launched InfoNgen®, a text analytics and sentiment analysis enterprise software product that uses artificial intelligence.
The company also productized TelescopeAI®, an artificial intelligence-based platform for IT operations and workforce management, which won a 2019 Big Innovation Award presented by the Business Intelligence Group.
In 2019, EPAM joined the Blockchain in Transport Alliance (BiTA). That year, the company launched EPAM Continuum, its service for business, experience and technology consulting.
The company also launched EPAM SolutionsHub, a catalogue of its software products, accelerators and open-source platforms. As part of its SolutionsHub launch, EPAM also released the Open-Source Contributor Index (OSCI), a tool that ranks the top open-source contributors by commercial organization.
In August 2021, EPAM expanded its presence in Latin America through the acquisition of Colombia-based S4N, a software development services firm specializing in the design and development of modern software products and enterprise platforms.
In July 2021, EPAM acquired CORE SE, a professional service provider specializing in IT strategy and technology-driven transformations, to further expand its Western European footprint in the DACH region.
In May 2021, EPAM acquired Just-BI, a Netherlands-based consultancy specializing SAP/S4HANA and enterprise data and analytics program management.
EPAM acquired Israel-based cyber-security services provider White-Hat Ltd. in May 2021.
In April 2021, EPAM acquired PolSource, a Salesforce Consulting Partner with more than 350 Salesforce specialists across the Americas and Europe.
In December 2021, EPAM joined the S&P 500.
In May 2021, EPAM Systems ranked 1,804 on the Forbes Global 2000 list.

CHAPTER 2
TECHNOLOGIES LEANT

Software Development Methodologies:

• High Level Overview: An overview of software development methodologies, their benefits, and their various types.

• Waterfall: An introduction to the traditional Waterfall methodology, which is a linear and sequential approach to software development.

• Agile: An overview of the Agile methodology, which emphasizes flexibility, customer collaboration, and incremental and iterative development.

• Scrum: An in-depth look at the Scrum framework, which is a popular Agile methodology that uses sprints, backlogs, and daily stand-ups to manage projects.

• Kanban: An introduction to the Kanban methodology, which emphasizes visual management, continuous flow, and limiting work in progress.

• Extreme Programming: An overview of Extreme Programming (XP), which is an Agile methodology that emphasizes customer involvement, continuous testing, and frequent releases.

• Test-Driven Development: An introduction to Test-Driven Development (TDD), which is a software development approach that focuses on creating automated tests before writing code.

• Behavior-Driven Development: An overview of Behavior-Driven Development (BDD), which is a software development approach that focuses on describing the behavior of a system in natural language.

• Summary: A summary of the main points of each methodology, their strengths and weaknesses, and how to choose the right methodology for your project

Version Control with GIT:

• Version control concept: Understanding the basics of version control, its benefits, and its various types.

• Download, install and configure GIT: Installing GIT on your local machine and configuring it with your user details and preferences.

• GitHub: Introduction to GitHub, a web-based hosting service for version control, and its features such as repositories, issues, and pull requests.

• Git graphical tools: Overview of graphical user interfaces (GUI) and Integrated Development Environments (IDEs) that can be used to work with Git.
• Git internals: Understanding the inner workings of Git, including how Git stores and manages versions of files, branches, commits, and merges.
• Undoing changes: How to undo changes made to files or the repository using Git commands such as revert, reset, and checkout.
• Branching and merge: Creating and managing branches in Git, and merging changes from one branch to another.
• Tags: Creating and managing tags in Git to mark specific points in the repository's history, such as release versions.
• Stash: How to use Git stash to temporarily save changes that are not yet ready to be committed.
• Remotes: Working with remote repositories in Git, such as cloning, pushing, and pulling changes from remote repositories.
• Branching strategies: Overview of different branching strategies and workflows such as Gitflow and Github Flow.

Software Testing Introduction:
• Introduction to Software Functional Testing: An overview of software functional testing, its importance, and the different types of functional testing.
• Test Planning: An overview of test planning, which involves identifying test objectives, test strategies, and test schedules.
• Requirements Testing: An introduction to requirements testing, which involves verifying that the software meets the specified requirements.
• Test Cases and Test Scenarios: An in-depth look at test cases and test scenarios, which are used to define the conditions under which software will be tested and the expected results.
• Defect Reporting: An overview of defect reporting, which involves identifying and documenting defects in the software.
• Test Results Reporting: An introduction to test results reporting, which involves analyzing and documenting the results of the software testing.
• Test Automation Basics: An overview of test automation, which involves using software tools to automate the testing process and reduce manual effort.

Java Basics:

• Data Types: An introduction to data types in Java, including primitive and reference types, and how to declare and use them.
• Conditions and Loops: An introduction to conditional statements, such as if-else statements and switch statements, and loops such as for loops and while loops.
• Arrays: An introduction to arrays in Java, including how to declare and initialize arrays, and how to access and modify their elements.
• Classes: An introduction to classes in Java, including how to declare and instantiate classes, and how to use them to create objects.
• Introduction to OOP: An introduction to object-oriented programming (OOP) concepts, such as encapsulation, inheritance, and polymorphism.
• Abstract Classes and Interfaces: An introduction to abstract classes and interfaces, which are used to define abstract types that can be implemented by other classes.
• Nested Classes: An introduction to nested classes in Java, including static nested classes, inner classes, and anonymous classes.
• Strings: An introduction to strings in Java, including how to create and manipulate string objects.
• Collections and Maps: An introduction to collections and maps in Java, including how to use them to store and manipulate groups of objects.
• Exceptions: An introduction to exceptions in Java, including how to use try-catch blocks to handle exceptions and how to create custom exceptions.
• Annotations: An introduction to annotations in Java, which are used to provide metadata about code elements.
• Generics: An introduction to generics in Java, which allow you to define classes and methods that can work with different types of objects.
• Enum: An introduction to enumerations in Java, which are used to define a fixed set of values.

• Wrapper Classes and Optional Classes: An introduction to wrapper classes, which are used to represent primitive data types as objects, and optional classes, which are used to represent values that may be null.
• Code Documentation: An introduction to code documentation in Java, including how to use Javadoc to document your code.

Data & Analytics - Introduction to SQL:

• Database Basics: An introduction to databases, their types, components, and architecture.

• SQL Foundation: An introduction to Structured Query Language (SQL), including how to create and manipulate tables, perform queries, and use aggregate functions.

• SQL for Analysis: An introduction to using SQL for data analysis, including how to filter, sort, and group data, and how to join tables

Clean Code - Introduction to Clean Code:

• Introduction to Clean Code: An overview of clean code and why it is important in software development.

• Writing Clean Functions: Best practices for writing clean functions, including how to make them small, focused, and testable.

• Naming: Best practices for naming code elements, including classes, methods, and variables.

• Comments: Best practices for writing comments, including when to use them and what to include.

• Error Handling: Best practices for error handling, including how to handle exceptions and write code that is resilient to errors.

Cloud Computing - Introduction to Cloud Computing:

• Introduction to Cloud Computing: An overview of cloud computing and its benefits.

• Cloud Deployment Models: An introduction to different cloud deployment models, including public, private, and hybrid clouds.

• Cloud Service Models: An introduction to different cloud service models, including infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS).

• Cloud Providers: An introduction to different cloud providers, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).

• Cloud Security: An overview of cloud security, including best practices for securing cloud-based applications and data.

Automated Testing Basics + Java:

OO Design Principles & Patterns

• Patterns in General: An introduction to different types of design patterns, including structural, behavioral, and creational patterns.

• Factory Pattern: A creational pattern that provides a way to create objects without specifying their concrete classes.

• Strategy Pattern: A behavioral pattern that enables the selection of an algorithm at runtime.

• Builder Pattern: A creational pattern that separates the construction of a complex object from its representation.

• Singleton Pattern: A creational pattern that ensures a class has only one instance and provides a global point of access to it.

Introduction to Test Automation and xUnit Test Framework:

• Introduction to Test Automation: An overview of test automation and its benefits.

• UI/API/Performance/Security/Mobile Testing: An introduction to different types of test automation, including user interface (UI), application programming interface (API), performance, security, and mobile testing.

• Build Tools (Maven): An introduction to Maven, a build automation tool used primarily for Java projects.

• TestNG: An xUnit test framework for Java that supports parameterized, data-driven, and parallel testing.

API Automation:

• Client-Server Architecture: An overview of client-server architecture, which is the basis of most web applications.

• HTTP and HTTP Request/Response: An introduction to Hypertext Transfer Protocol (HTTP) and HTTP request/response messages.

• JSON and XML: An introduction to JavaScript Object Notation (JSON) and Extensible Markup Language (XML), two commonly used data formats for web applications.

• Postman: A popular tool for testing and debugging API requests.

Selenium WebDriver (Basic+Advanced):

• Introduction: An overview of Selenium WebDriver, a popular web testing tool.

• HTML and CSS: An introduction to HTML and CSS, the building blocks of web pages.
• XPath: A language used for selecting nodes in an XML or HTML document.
• Selenium WebDriver: An introduction to the WebDriver API and its methods for interacting with web pages.
• WebDriver Waiters: An introduction to the wait methods in WebDriver, which can help synchronize tests with the page under test.
• JS Executor: An advanced feature of WebDriver that allows JavaScript code to be executed within the context of a web page.

Automation Framework:
• Automation Framework: Maven + xUnit + WebDriver: This involves the use of Maven as a build tool, xUnit as the testing framework, and WebDriver for automating the web application. This helps in creating a structured and scalable test automation framework.
• Page Object: This is a design pattern that separates the page elements of a web application from the test scripts, making the tests more maintainable and reducing code duplication.
• Page Factory: This is an extension of the Page Object pattern that uses annotations to initialize web elements, making the code more readable and maintainable.
• Singleton: This is a design pattern that ensures only one instance of a class is created, which can be useful in maintaining the state of the application during the test.
• Production AT Framework: This includes various approaches like TDD (Test Driven Development), KDT (Keyword Driven Testing), DDT (Data Driven Testing), DDD (Domain Driven Design), BDD (Behavior Driven Development), and BDD + Cucumber. These approaches help in creating a robust and maintainable test automation framework that aligns with the development process.
• ATF Architecture: This is an architecture that helps in structuring the test automation framework and integrating it with the development process.
• Continuous Integration with Jenkins: This involves integrating the test automation framework with Jenkins, a popular continuous integration tool, to run the tests automatically and generate reports. This helps in achieving continuous testing and faster feedback.

CHAPTER 3
PROJECTS UNDERTAKEN

3.1 Amazon-UI Test

For my project report, I conducted a UI test for Amazon, focusing on the login process, wishlist creation, adding products, removing products, and deleting both products and wishlist.

1. Login Process: I first tested the login process of Amazon, making sure that users can enter their username and password correctly and access their account. I also checked for any error messages or bugs that may occur during the login process.
2. Wishlist Creation: I then proceeded to test the wishlist creation feature of Amazon. I created a new wishlist and ensured that the name was entered correctly.
3. Searching Products: I tested the ability to search for a product in the search bar and ensured that the product name was entered correctly.
4. Adding Products: Next, I tested the ability to add products to a wishlist. I selected a product and added it to the newly created wishlist. I then checked that the product was added correctly and appeared in the wishlist.
5. Deletion of Product: I tested the ability to delete product form wishlist. I removed a product from the wishlist and checked that it was no longer in the wishlist.
6. Renaming of Wishlist: After deletion of product, I have renamed wishlist to new name.
7. Deletion of Wishlist: Finally, I tested the ability to delete wishlist. I deleted the wishlist entirely and ensured that it was removed from the user's account.

Overall, I found that the UI of Amazon was intuitive and easy to use. The login process was straightforward, and wishlist creation and management were simple. Searching products and adding products, deleting products and wishlist were also easy to do. There were no significant issues or bugs that I encountered during the test.

3.1.1 Technologies Used

For my project report, I used several technologies to perform UI testing for the Amazon application. These technologies include Maven, Java, Selenium Web--Driver, Jenkins, TestNG, and the Page Object Pattern.

1. Maven: Maven is a build automation tool that helps in managing project dependencies, build processes, and deployment. It simplifies the development process by automating the building of projects and managing libraries and dependencies. I used Maven to build and manage the project dependencies of my UI testing framework.

2. Java: Java is a popular programming language used for developing enterprise applications, web applications, and mobile applications. I used Java to write the test scripts for my UI testing framework. Java is a robust language with an extensive library of APIs and tools, making it an excellent choice for UI testing.

3. Selenium WebDriver: Selenium WebDriver is a popular open-source tool used for automating web browsers. It supports different browsers, including Chrome, Firefox, and Safari, and provides a set of APIs for interacting with web elements. I used Selenium WebDriver to automate my UI testing for the Amazon application.

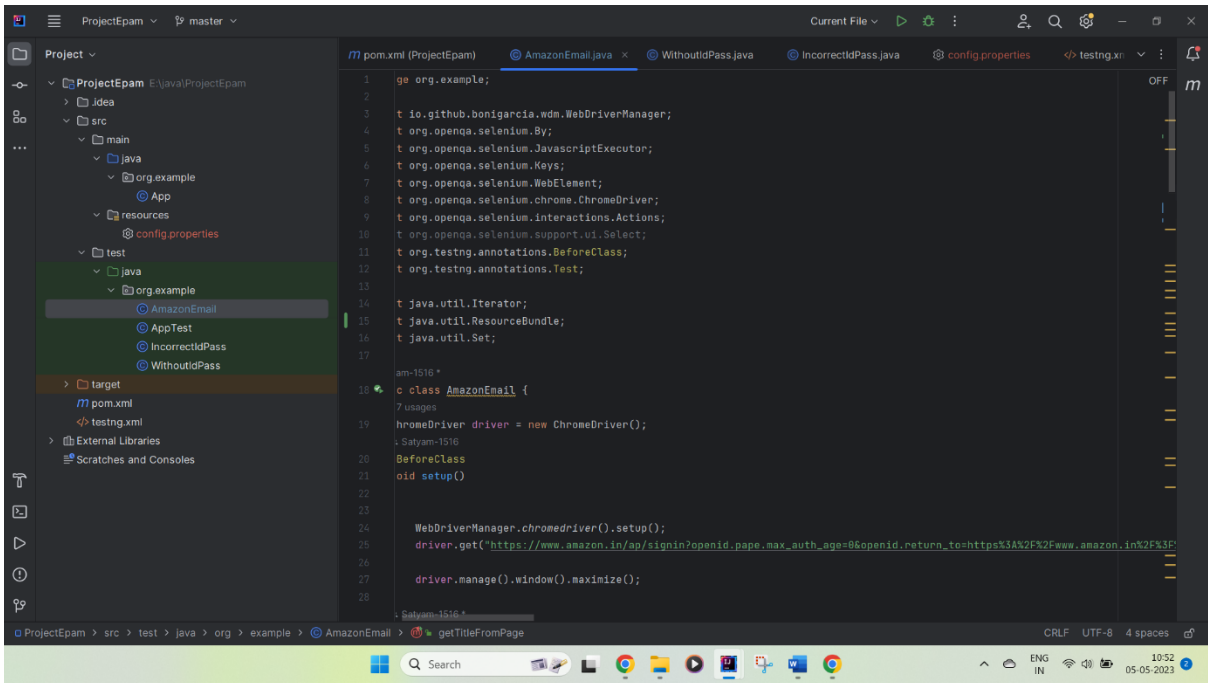
4. Jenkins: Jenkins is an open-source automation server that helps in building, testing, and deploying software applications. It provides a wide range of plugins that enable continuous integration and continuous deployment. I used Jenkins to set up and run automated tests as part of the CI/CD pipeline.

5. TestNG: TestNG is a testing framework that provides a wide range of features for automating unit tests, integration tests, and functional tests. It supports different test types, including data-driven tests, parameterized tests, and dependency tests. I used TestNG to create and execute test cases for my UI testing framework.

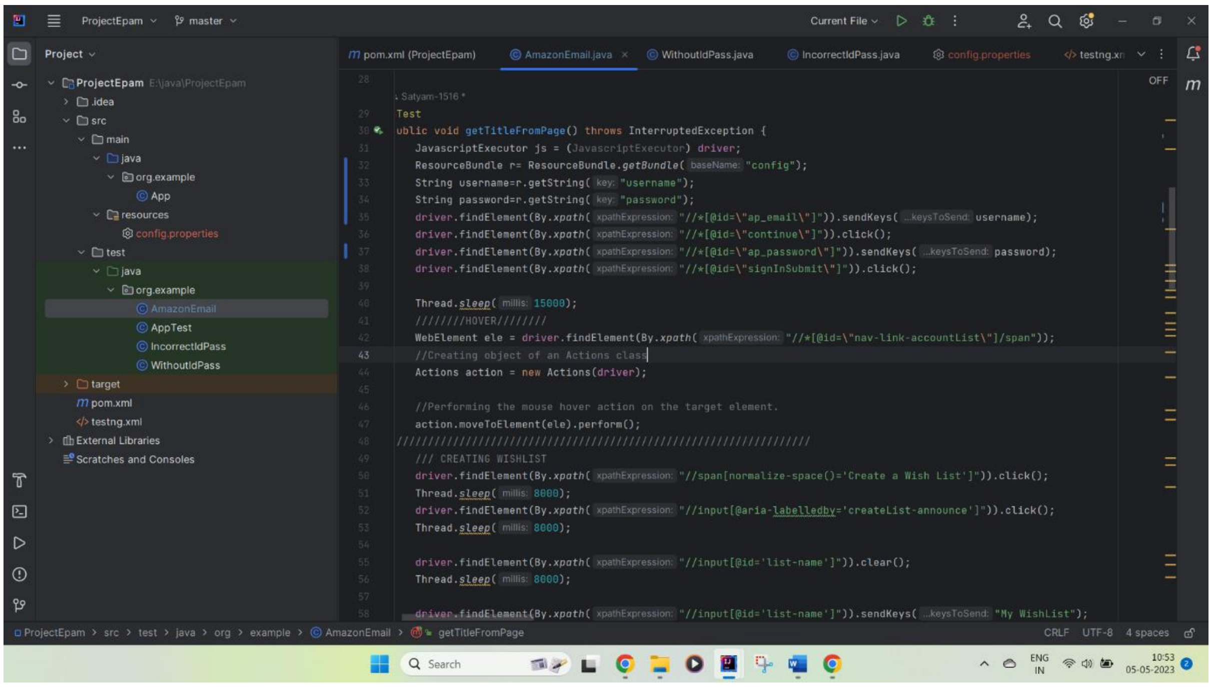
6. Page Object Pattern: The Page Object Pattern is a design pattern used for implementing UI testing frameworks. It helps in creating a modular and maintainable code structure by separating the page objects from the test scripts. I used the Page Object Pattern to structure my UI testing framework, making it easier to maintain and update.

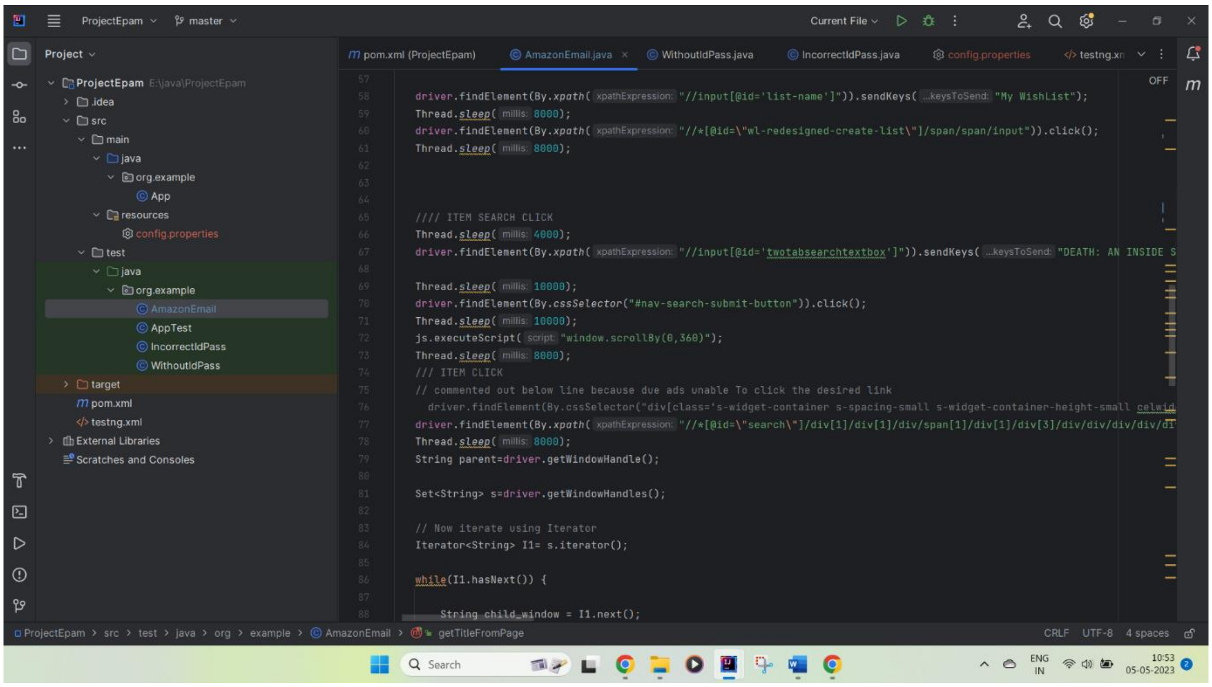
In summary, I used Maven, Java, Selenium WebDriver, Jenkins, TestNG, and the Page Object Pattern to create a robust and reliable UI testing framework for the Amazon application. These technologies helped me automate the testing process, reducing the time and effort required to test the application manually.

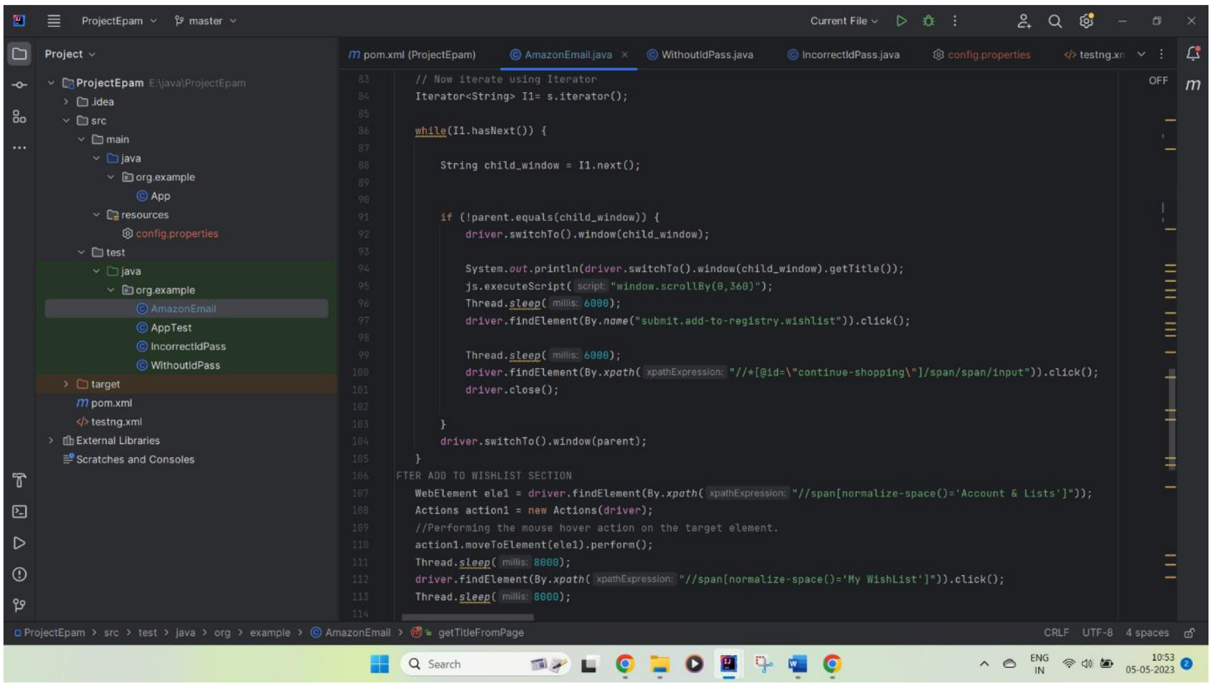
3.1.2 Snapshots



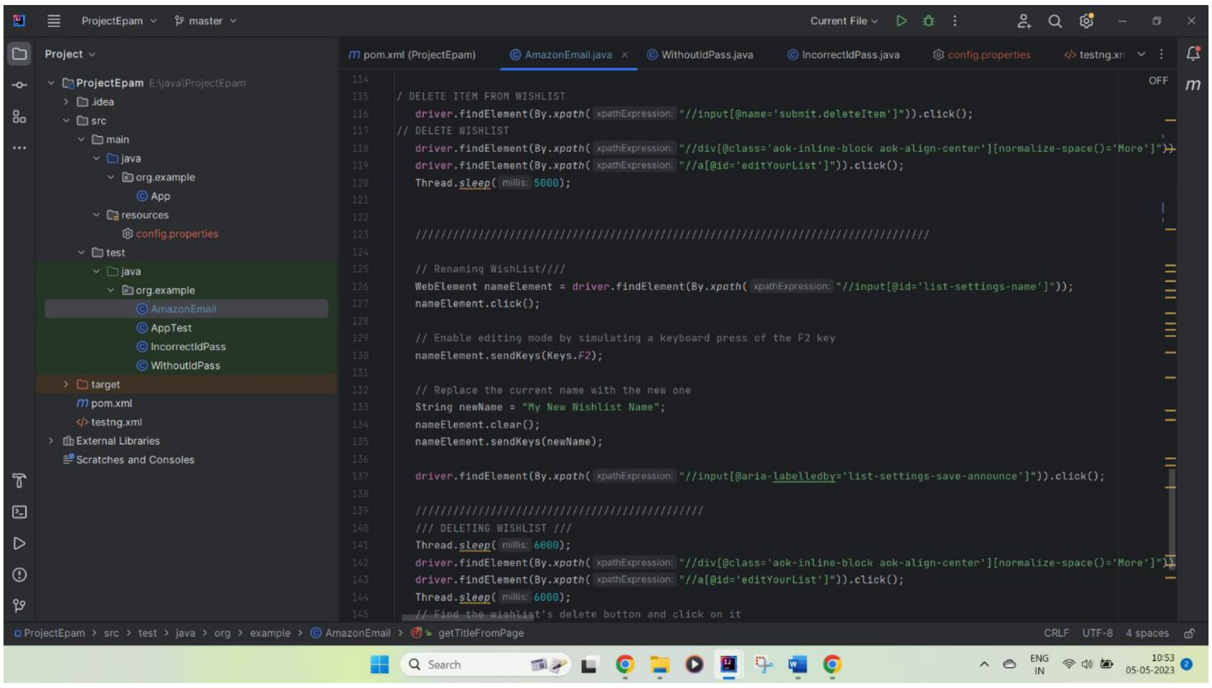
```
ge org.example;

t io.github.bonigarcia.wdm.WebDriverManager;
t org.openqa.selenium.By;
t org.openqa.selenium.JavascriptExecutor;
t org.openqa.selenium.Keys;
t org.openqa.selenium.WebElement;
t org.openqa.selenium.chrome.ChromeDriver;
t org.openqa.selenium.interactions.Actions;
t org.openqa.selenium.support.ui.Select;
t org.testng.annotations.BeforeClass;
t org.testng.annotations.Test;

t java.util.Iterator;
t java.util.ResourceBundle;
t java.util.Set;

am-1516 *
c class AmazonEmail {
7 usages
hromeDriver driver = new ChromeDriver();
. Satyam-1516
BeforeClass
oid setup()


   WebDriverManager.chromedriver().setup();
   driver.get("https://www.amazon.in/ap/signin?openid.pape.max_auth_age=0&openid.return_to=https%3A%2F%2Fwww.amazon.in%2F%3F

   driver.manage().window().maximize();

. Satyam-1516 *
```


```
Satyam-1516 *
Test
ublic void getTitleFromPage() throws InterruptedException {
    JavascriptExecutor js = (JavascriptExecutor) driver;
    ResourceBundle r= ResourceBundle.getBundle("config");
    String username=r.getString("username");
    String password=r.getString("password");
    driver.findElement(By.xpath("//*[@id=\"ap_email\"]")).sendKeys(username);
    driver.findElement(By.xpath("//*[@id=\"continue\"]")).click();
    driver.findElement(By.xpath("//*[@id=\"ap_password\"]")).sendKeys(password);
    driver.findElement(By.xpath("//*[@id=\"signInSubmit\"]")).click();

    Thread.sleep(15000);
    ////////HOVER////////
    WebElement ele = driver.findElement(By.xpath("//*[@id=\"nav-link-accountList\"]/span"));
    //Creating object of an Actions class
    Actions action = new Actions(driver);

    //Performing the mouse hover action on the target element.
    action.moveToElement(ele).perform();
////////////////////////////////////////////////////////////////////
    /// CREATING WISHLIST
    driver.findElement(By.xpath("//span[normalize-space()='Create a Wish List']")).click();
    Thread.sleep(8000);
    driver.findElement(By.xpath("//input[@aria-labelledby='createList-announce']")).click();
    Thread.sleep(8000);

    driver.findElement(By.xpath("//input[@id='list-name']")).clear();
    Thread.sleep(8000);

    driver.findElement(By.xpath("//input[@id='list-name']")).sendKeys("My WishList");
```

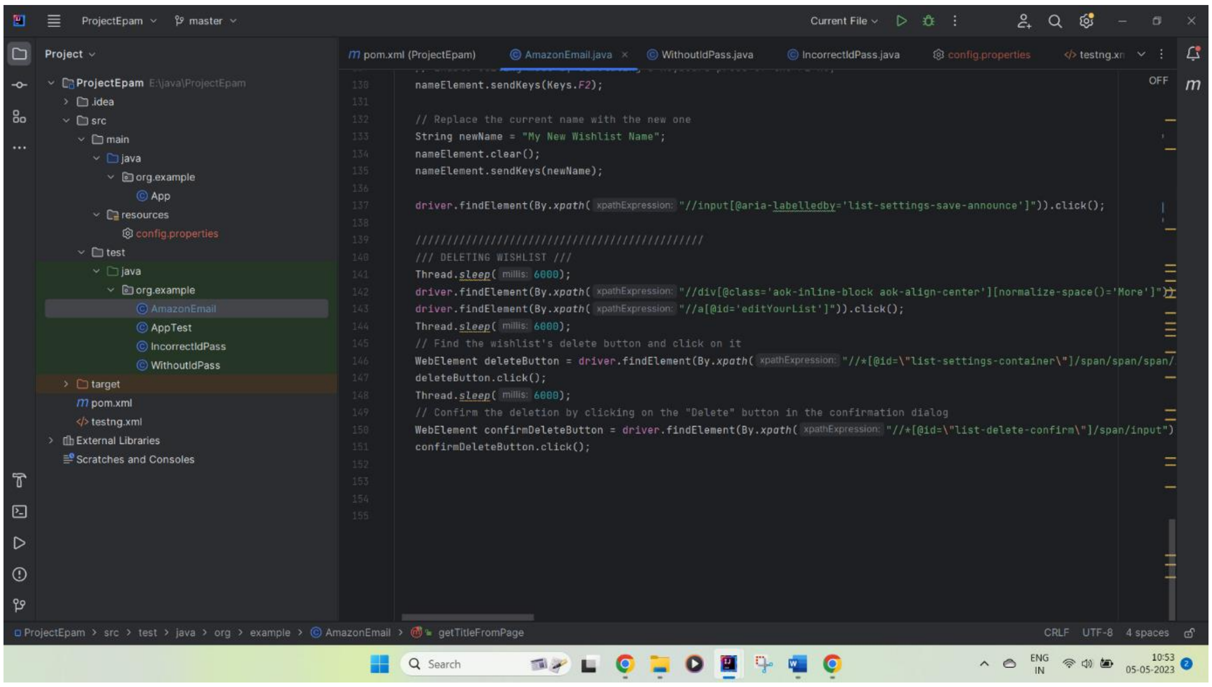

```

    driver.findElement(By.xpath("//input[@id='list-name']")).sendKeys("My WishList");
    Thread.sleep(8000);
    driver.findElement(By.xpath("//*[@id=\"wl-redesigned-create-list\"]/span/span/input")).click();
    Thread.sleep(8000);



    //// ITEM SEARCH CLICK
    Thread.sleep(4000);
    driver.findElement(By.xpath("//input[@id='twotabsearchtextbox']")).sendKeys("DEATH: AN INSIDE S

    Thread.sleep(10000);
    driver.findElement(By.cssSelector("#nav-search-submit-button")).click();
    Thread.sleep(10000);
    js.executeScript("window.scrollBy(0,360)");
    Thread.sleep(8000);
    /// ITEM CLICK
    // commented out below line because due ads unable To click the desired link
      driver.findElement(By.cssSelector("div[class='s-widget-container s-spacing-small s-widget-container-height-small celwid
    driver.findElement(By.xpath("//*[@id=\"search\"]/div[1]/div[1]/div/span[1]/div[1]/div[3]/div/div/div/div/di
    Thread.sleep(8000);
    String parent=driver.getWindowHandle();

    Set<String> s=driver.getWindowHandles();

    // Now iterate using Iterator
    Iterator<String> I1= s.iterator();

    while(I1.hasNext()) {

        String child_window = I1.next();
```



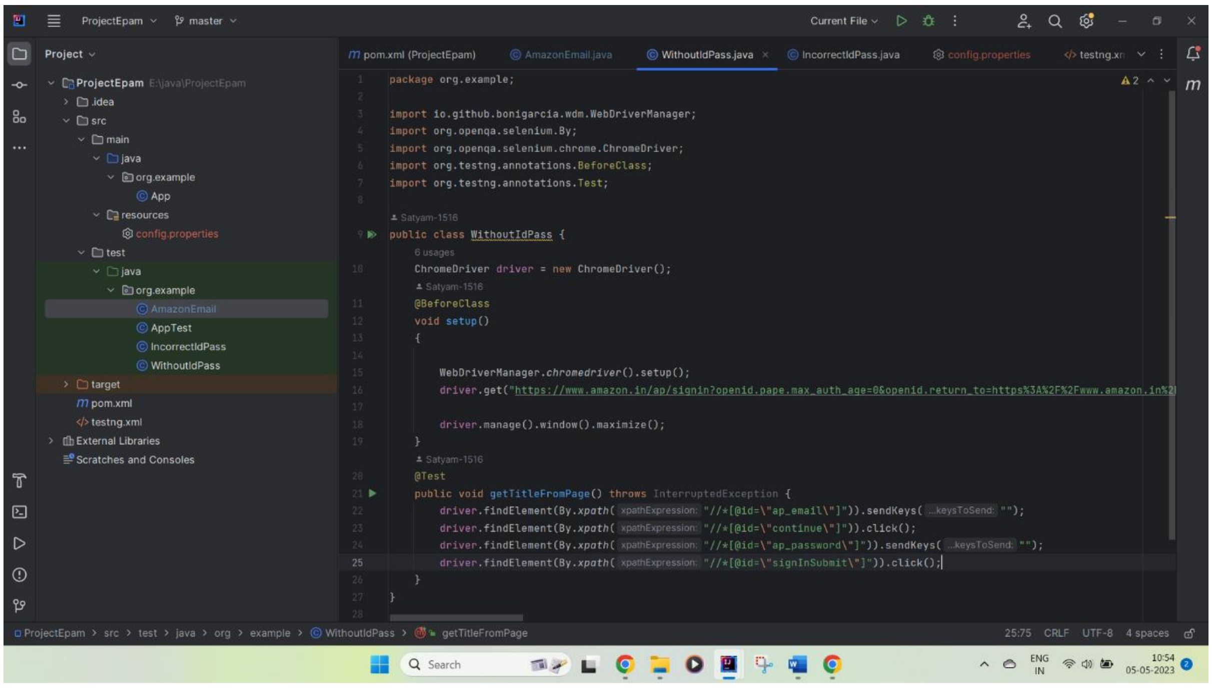
```java
    // Now iterate using Iterator
    Iterator<String> I1= s.iterator();

    while(I1.hasNext()) {

        String child_window = I1.next();


        if (!parent.equals(child_window)) {
            driver.switchTo().window(child_window);

            System.out.println(driver.switchTo().window(child_window).getTitle());
            js.executeScript("window.scrollBy(0,360)");
            Thread.sleep(6000);
            driver.findElement(By.name("submit.add-to-registry.wishlist")).click();

            Thread.sleep(6000);
            driver.findElement(By.xpath("//*[@id=\"continue-shopping\"]/span/span/input")).click();
            driver.close();

        }
        driver.switchTo().window(parent);
    }
FTER ADD TO WISHLIST SECTION
    WebElement ele1 = driver.findElement(By.xpath("//span[normalize-space()='Account & Lists']"));
    Actions action1 = new Actions(driver);
    //Performing the mouse hover action on the target element.
    action1.moveToElement(ele1).perform();
    Thread.sleep(8000);
    driver.findElement(By.xpath("//span[normalize-space()='My WishList']")).click();
    Thread.sleep(8000);
```



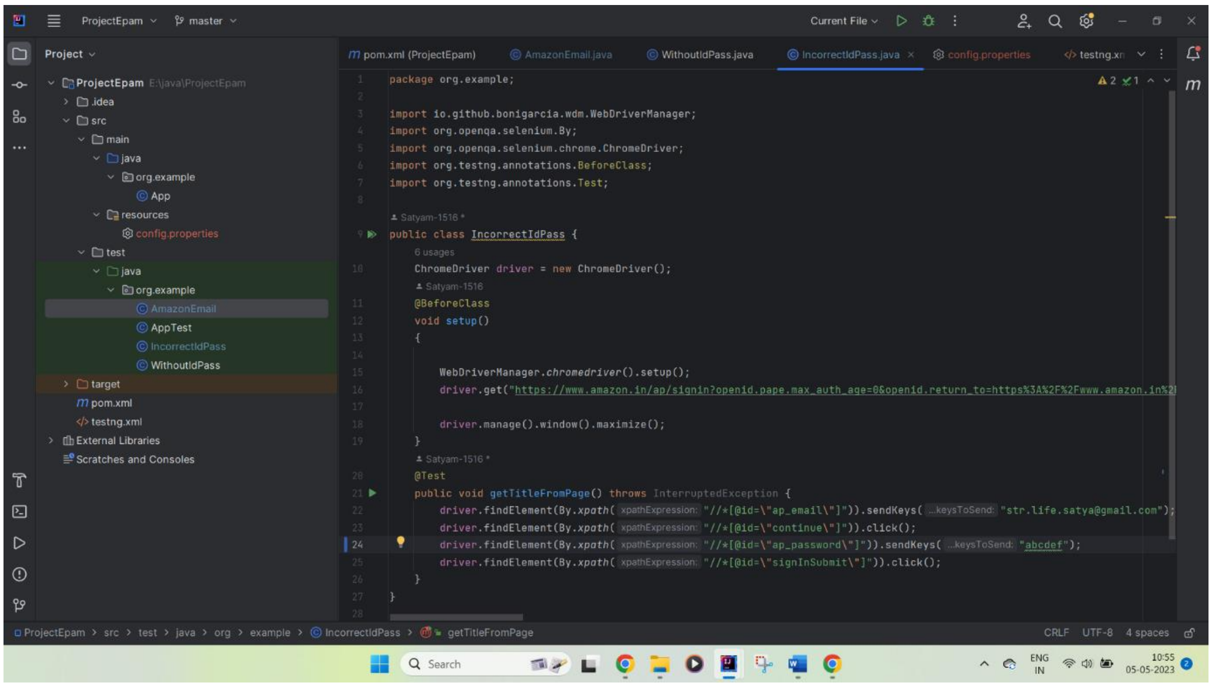
```java
/ DELETE ITEM FROM WISHLIST
    driver.findElement(By.xpath("//input[@name='submit.deleteItem']")).click();
// DELETE WISHLIST
    driver.findElement(By.xpath("//div[@class='aok-inline-block aok-align-center'][normalize-space()='More']")
    driver.findElement(By.xpath("//a[@id='editYourList']")).click();
    Thread.sleep(5000);


    ///////////////////////////////////////////////////////////////////////////////

    // Renaming WishList////
    WebElement nameElement = driver.findElement(By.xpath("//input[@id='list-settings-name']"));
    nameElement.click();

    // Enable editing mode by simulating a keyboard press of the F2 key
    nameElement.sendKeys(Keys.F2);

    // Replace the current name with the new one
    String newName = "My New Wishlist Name";
    nameElement.clear();
    nameElement.sendKeys(newName);

    driver.findElement(By.xpath("//input[@aria-labelledby='list-settings-save-announce']")).click();

    ///////////////////////////////////////////////
    /// DELETING WISHLIST ///
    Thread.sleep(6000);
    driver.findElement(By.xpath("//div[@class='aok-inline-block aok-align-center'][normalize-space()='More']")
    driver.findElement(By.xpath("//a[@id='editYourList']")).click();
    Thread.sleep(6000);
    // Find the wishlist's delete button and click on it
```



```java
    nameElement.sendKeys(Keys.F2);

    // Replace the current name with the new one
    String newName = "My New Wishlist Name";
    nameElement.clear();
    nameElement.sendKeys(newName);

    driver.findElement(By.xpath("//input[@aria-labelledby='list-settings-save-announce']")).click();

    ///////////////////////////////////////////
    /// DELETING WISHLIST ///
    Thread.sleep(6000);
    driver.findElement(By.xpath("//div[@class='aok-inline-block aok-align-center'][normalize-space()='More']"));
    driver.findElement(By.xpath("//a[@id='editYourList']")).click();
    Thread.sleep(6000);
    // Find the wishlist's delete button and click on it
    WebElement deleteButton = driver.findElement(By.xpath("//*[@id=\"list-settings-container\"]/span/span/span/
    deleteButton.click();
    Thread.sleep(6000);
    // Confirm the deletion by clicking on the "Delete" button in the confirmation dialog
    WebElement confirmDeleteButton = driver.findElement(By.xpath("//*[@id=\"list-delete-confirm\"]/span/input")
    confirmDeleteButton.click();
```



```java
package org.example;

import io.github.bonigarcia.wdm.WebDriverManager;
import org.openqa.selenium.By;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;

public class WithoutIdPass {
    ChromeDriver driver = new ChromeDriver();
    @BeforeClass
    void setup()
    {

        WebDriverManager.chromedriver().setup();
        driver.get("https://www.amazon.in/ap/signin?openid.pape.max_auth_age=0&openid.return_to=https%3A%2F%2Fwww.amazon.in%2

        driver.manage().window().maximize();
    }
    @Test
    public void getTitleFromPage() throws InterruptedException {
        driver.findElement(By.xpath("//*[@id=\"ap_email\"]")).sendKeys("");
        driver.findElement(By.xpath("//*[@id=\"continue\"]")).click();
        driver.findElement(By.xpath("//*[@id=\"ap_password\"]")).sendKeys("");
        driver.findElement(By.xpath("//*[@id=\"signInSubmit\"]")).click();
    }
}
```



```java
package org.example;

import io.github.bonigarcia.wdm.WebDriverManager;
import org.openqa.selenium.By;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;

public class IncorrectIdPass {
    ChromeDriver driver = new ChromeDriver();
    @BeforeClass
    void setup()
    {

        WebDriverManager.chromedriver().setup();
        driver.get("https://www.amazon.in/ap/signin?openid.pape.max_auth_age=0&openid.return_to=https%3A%2F%2Fwww.amazon.in%2

        driver.manage().window().maximize();
    }
    @Test
    public void getTitleFromPage() throws InterruptedException {
        driver.findElement(By.xpath("//*[@id=\"ap_email\"]")).sendKeys("str.life.satya@gmail.com");
        driver.findElement(By.xpath("//*[@id=\"continue\"]")).click();
        driver.findElement(By.xpath("//*[@id=\"ap_password\"]")).sendKeys("abcdef");
        driver.findElement(By.xpath("//*[@id=\"signInSubmit\"]")).click();
    }
}
```

3.2 Individual Tasks

1. The project focused on automated testing, utilizing various technologies such as Maven, Java, Selenium WebDriver, TestNG, and Jenkins.

2. Several tasks were completed as part of the project, including installing Maven, downloading a test project from GitHub, and changing the Junit version.

3. The main objective was to create automated tests for website functionality, such as creating new paste and using the Google Cloud Platform Pricing Calculator.

4. A framework was developed to facilitate the automation of tests, including WebDriver management for browser connectors, Page Object/Page Factory for page abstractions, Models for business objects of the required elements, and Property files with test data for at least two different environments.

5. The framework also included XML suites for smoke tests and other tests, an option for running with Jenkins and browser parameterization, test suite, environment, and screenshot capture in case of test failure.

6. Jenkins was used to set up continuous integration, which involved creating a task to clone the project and launch tests from the project in the Java directory using the mvn test goal. Build triggers were set up to perform the task every 5 minutes.

7. The overall goal of the project was to demonstrate proficiency in automated testing using various technologies and tools commonly used in industry settings.

3.2.1 Technologies Used

For my project report, I used several technologies to perform UI testing for the Amazon application. These technologies include Maven, Java, Selenium Web--Driver, Jenkins, TestNG, and the Page Object Pattern.

1. Maven: Maven is a build automation tool that helps in managing project dependencies, build processes, and deployment. It simplifies the development process by automating the building of projects and managing libraries and dependencies. I used Maven to build and manage the project dependencies of my UI testing framework.

2. Java: Java is a popular programming language used for developing enterprise applications, web applications, and mobile applications. I used Java to write the test scripts for my UI testing framework. Java is a robust language with an extensive library of APIs and tools, making it an excellent choice for UI testing.

3. Selenium WebDriver: Selenium WebDriver is a popular open-source tool used for automating web browsers. It supports different browsers, including Chrome, Firefox, and Safari, and provides a set of APIs for interacting with web elements. I used Selenium WebDriver to automate my UI testing for the Amazon application.

4. Jenkins: Jenkins is an open-source automation server that helps in building, testing, and deploying software applications. It provides a wide range of plugins that enable continuous integration and continuous deployment. I used Jenkins to set up and run automated tests as part of the CI/CD pipeline.

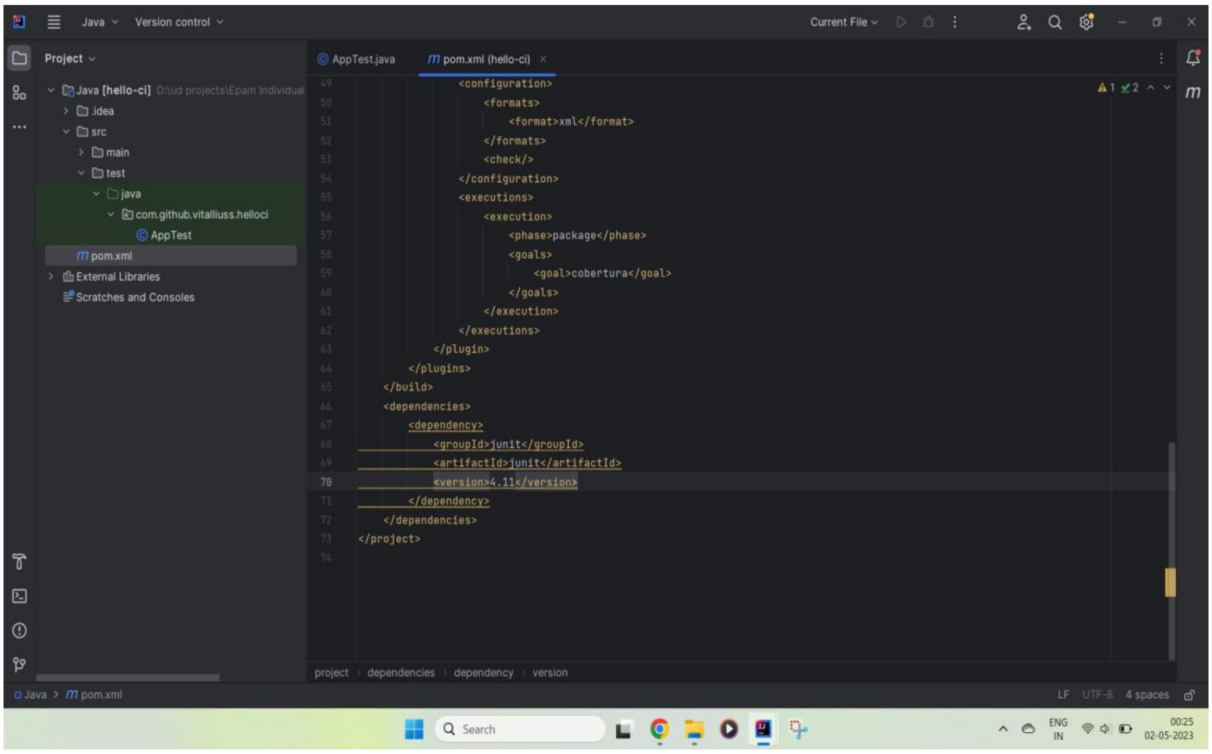
5. TestNG: TestNG is a testing framework that provides a wide range of features for automating unit tests, integration tests, and functional tests. It supports different test types, including data-driven tests, parameterized tests, and dependency tests. I used TestNG to create and execute test cases for my UI testing framework.

6. Page Object Pattern: The Page Object Pattern is a design pattern used for implementing UI testing frameworks. It helps in creating a modular and maintainable code structure by separating the page objects from the test scripts. I used the Page Object Pattern to structure my UI testing framework, making it easier to maintain and update.

In summary, I used Maven, Java, Selenium WebDriver, Jenkins, TestNG, and the Page Object Pattern to create a robust and reliable UI testing framework for the Amazon application. These technologies helped me automate the testing process, reducing the time and effort required to test the application manually.

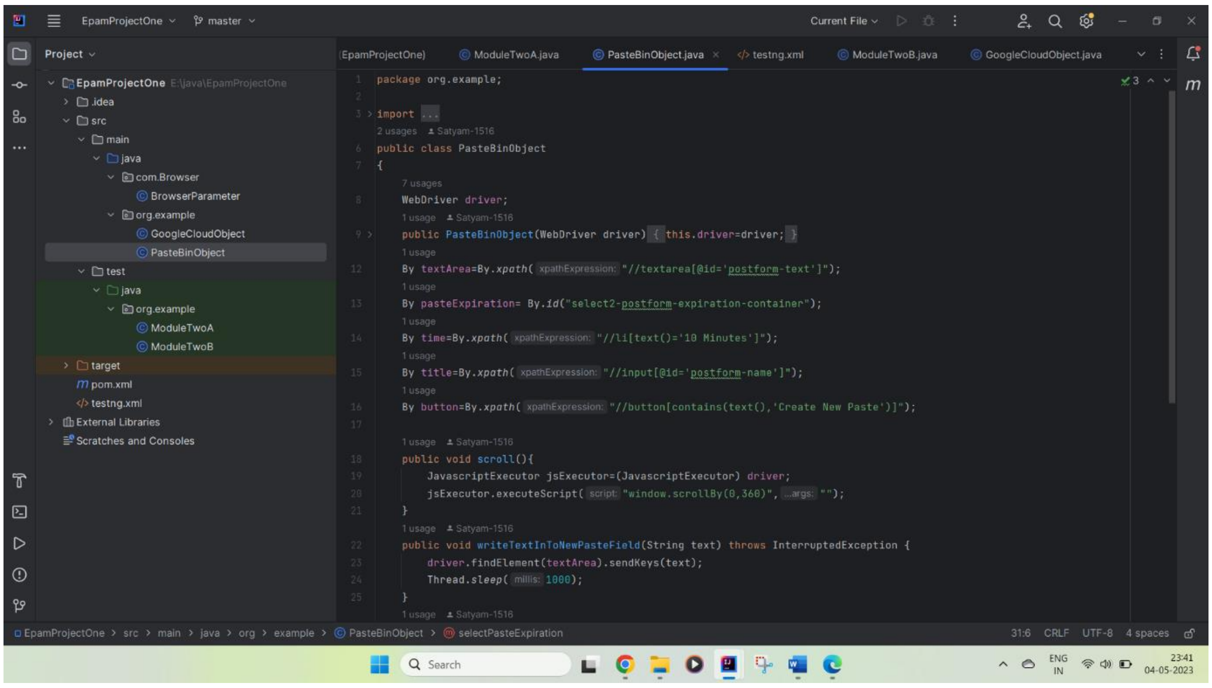
3.2.2 Snapshots

Changed the junit version in the file pom.xml from 4.12 to 4.11.

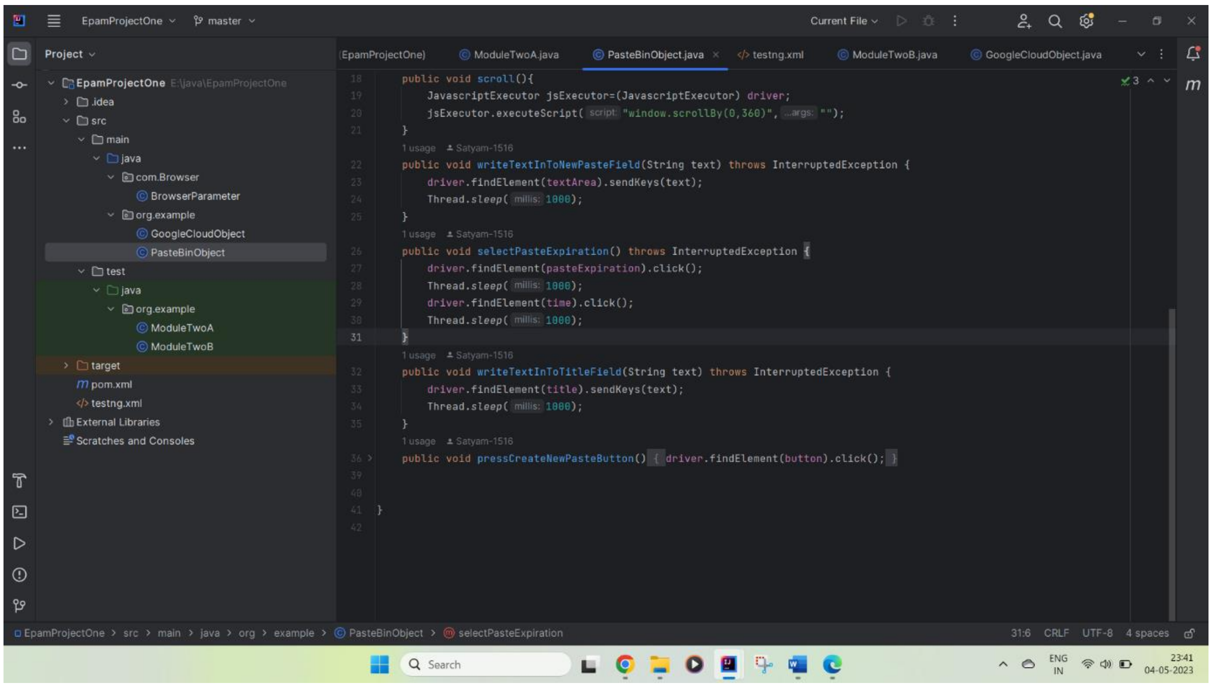


```
                    <configuration>
                        <formats>
                            <format>xml</format>
                        </formats>
                        <check/>
                    </configuration>
                    <executions>
                        <execution>
                            <phase>package</phase>
                            <goals>
                                <goal>cobertura</goal>
                            </goals>
                        </execution>
                    </executions>
                </plugin>
            </plugins>
        </build>
        <dependencies>
            <dependency>
                <groupId>junit</groupId>
                <artifactId>junit</artifactId>
                <version>4.11</version>
            </dependency>
        </dependencies>
</project>
```

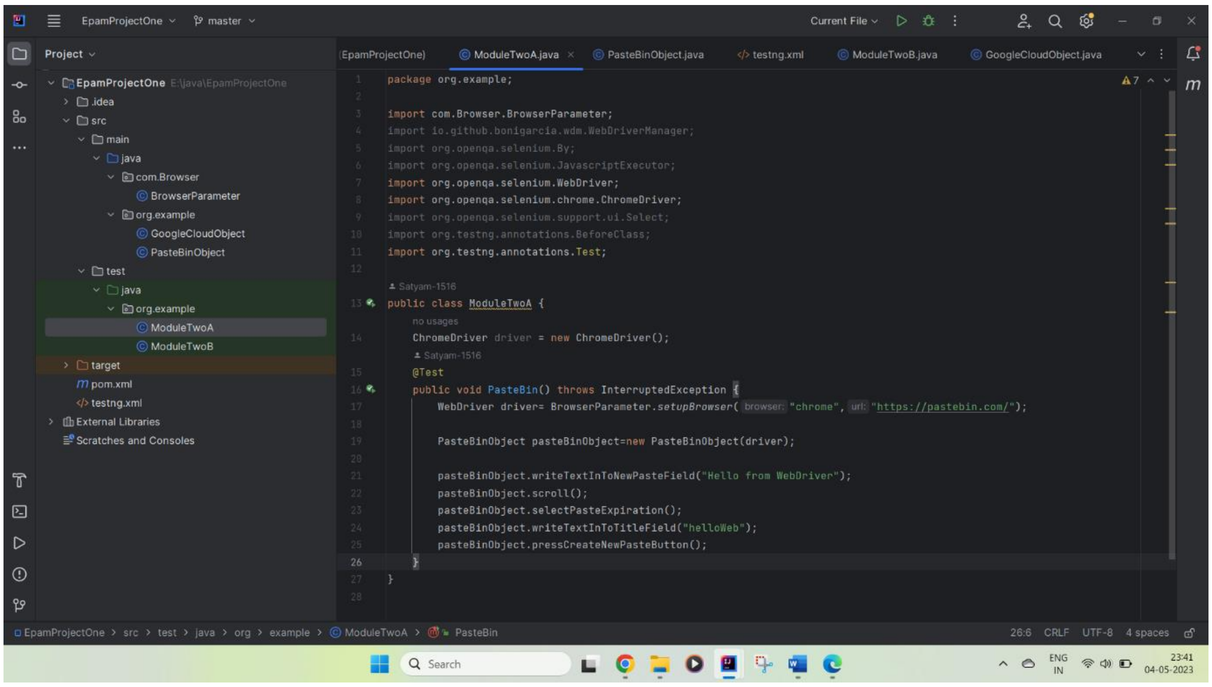
Opened https://pastebin.com/ and created 'New Paste' with the following attributes: - Code: "Hello from WebDriver" - Paste Expiration: "10 Minutes" - Paste Name / Title: "helloweb" using Selenium WebDriver.



```java
package org.example;

import ...
public class PasteBinObject
{
    WebDriver driver;
    public PasteBinObject(WebDriver driver) { this.driver=driver; }
    By textArea=By.xpath("//textarea[@id='postform-text']");
    By pasteExpiration= By.id("select2-postform-expiration-container");
    By time=By.xpath("//li[text()='10 Minutes']");
    By title=By.xpath("//input[@id='postform-name']");
    By button=By.xpath("//button[contains(text(),'Create New Paste')]");

    public void scroll(){
        JavascriptExecutor jsExecutor=(JavascriptExecutor) driver;
        jsExecutor.executeScript("window.scrollBy(0,360)", "");
    }
    public void writeTextInToNewPasteField(String text) throws InterruptedException {
        driver.findElement(textArea).sendKeys(text);
        Thread.sleep(1000);
    }
```

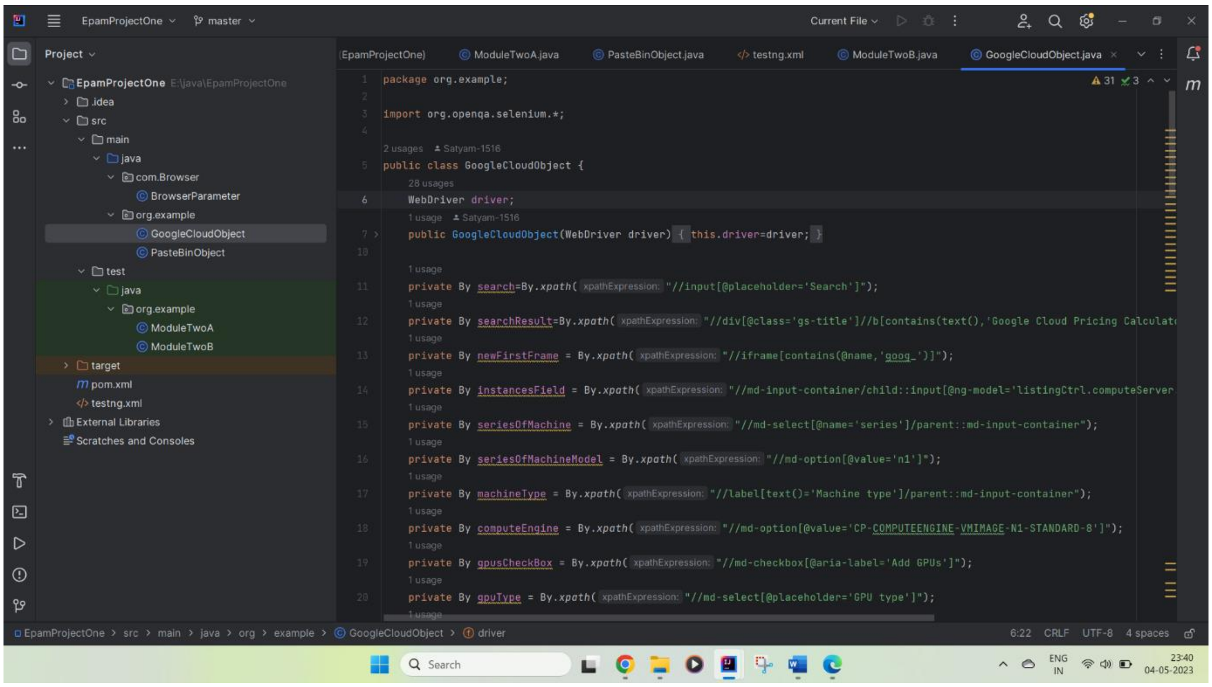


```java
    public void scroll(){
        JavascriptExecutor jsExecutor=(JavascriptExecutor) driver;
        jsExecutor.executeScript("window.scrollBy(0,360)", "");
    }
    public void writeTextInToNewPasteField(String text) throws InterruptedException {
        driver.findElement(textArea).sendKeys(text);
        Thread.sleep(1000);
    }
    public void selectPasteExpiration() throws InterruptedException {
        driver.findElement(pasteExpiration).click();
        Thread.sleep(1000);
        driver.findElement(time).click();
        Thread.sleep(1000);
    }
    public void writeTextInToTitleField(String text) throws InterruptedException {
        driver.findElement(title).sendKeys(text);
        Thread.sleep(1000);
    }
    public void pressCreateNewPasteButton() { driver.findElement(button).click(); }


}
```

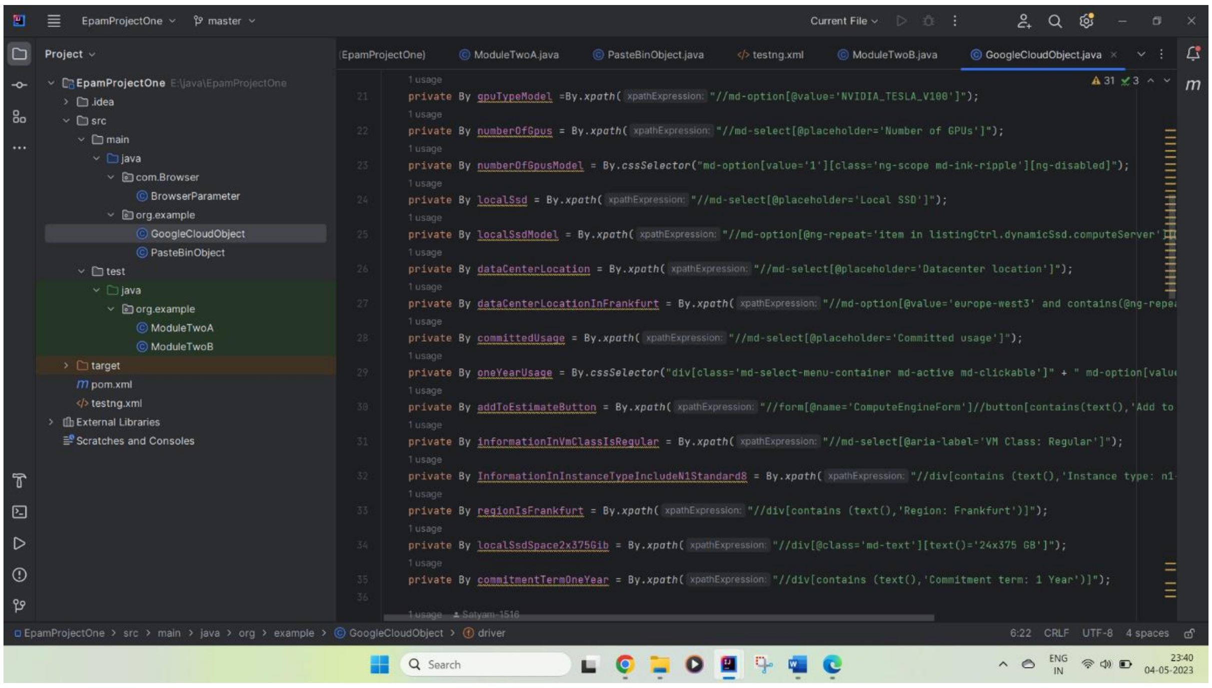


```java
package org.example;

import com.Browser.BrowserParameter;
import io.github.bonigarcia.wdm.WebDriverManager;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;

public class ModuleTwoA {
    ChromeDriver driver = new ChromeDriver();
    @Test
    public void PasteBin() throws InterruptedException {
        WebDriver driver= BrowserParameter.setupBrowser( browser: "chrome", url: "https://pastebin.com/");

        PasteBinObject pasteBinObject=new PasteBinObject(driver);

        pasteBinObject.writeTextInToNewPasteField("Hello from WebDriver");
        pasteBinObject.scroll();
        pasteBinObject.selectPasteExpiration();
        pasteBinObject.writeTextInToTitleField("helloWeb");
        pasteBinObject.pressCreateNewPasteButton();
    }
}
```

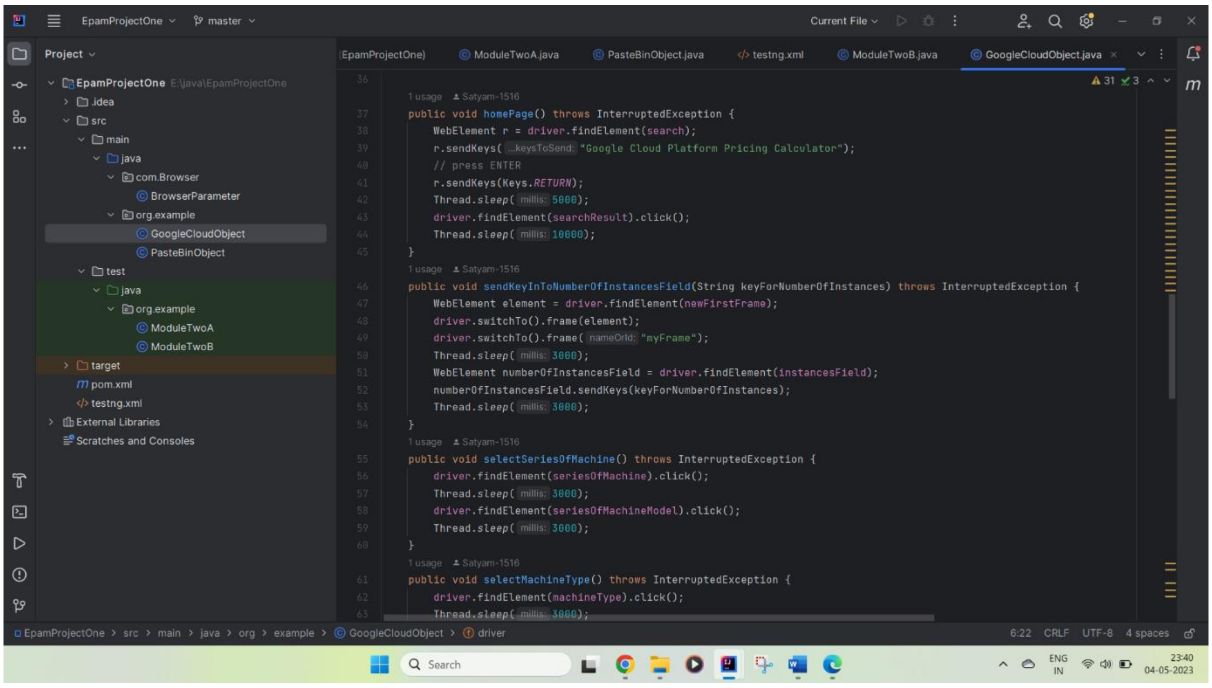
Opened https://cloud.google.com/ and calculated monthly rent using Selenium WebDriver.



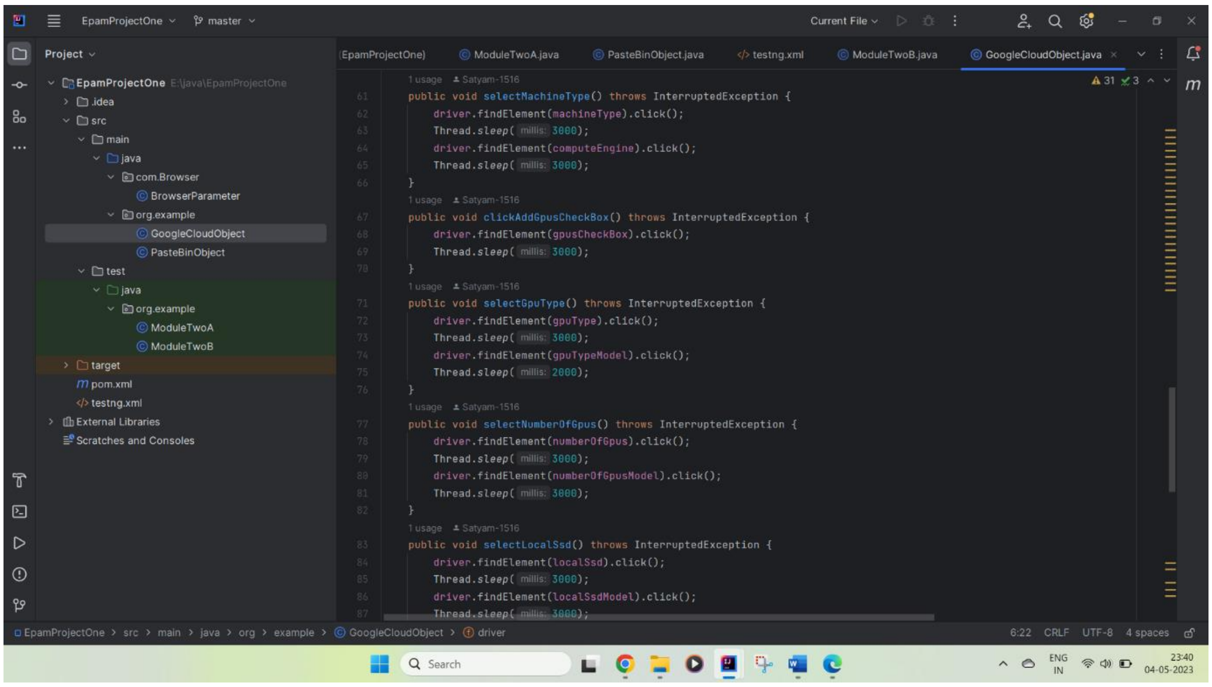
```java
package org.example;

import org.openqa.selenium.*;

public class GoogleCloudObject {
    WebDriver driver;
    public GoogleCloudObject(WebDriver driver) { this.driver=driver; }

    private By search=By.xpath( xpathExpression: "//input[@placeholder='Search']");
    private By searchResult=By.xpath( xpathExpression: "//div[@class='gs-title']//b[contains(text(),'Google Cloud Pricing Calculat
    private By newFirstFrame = By.xpath( xpathExpression: "//iframe[contains(@name,'goog_')]");
    private By instancesField = By.xpath( xpathExpression: "//md-input-container/child::input[@ng-model='listingCtrl.computeServer
    private By seriesOfMachine = By.xpath( xpathExpression: "//md-select[@name='series']/parent::md-input-container");
    private By seriesOfMachineModel = By.xpath( xpathExpression: "//md-option[@value='n1']");
    private By machineType = By.xpath( xpathExpression: "//label[text()='Machine type']/parent::md-input-container");
    private By computeEngine = By.xpath( xpathExpression: "//md-option[@value='CP-COMPUTEENGINE-VMIMAGE-N1-STANDARD-8']");
    private By gpusCheckBox = By.xpath( xpathExpression: "//md-checkbox[@aria-label='Add GPUs']");
    private By gpuType = By.xpath( xpathExpression: "//md-select[@placeholder='GPU type']");
```



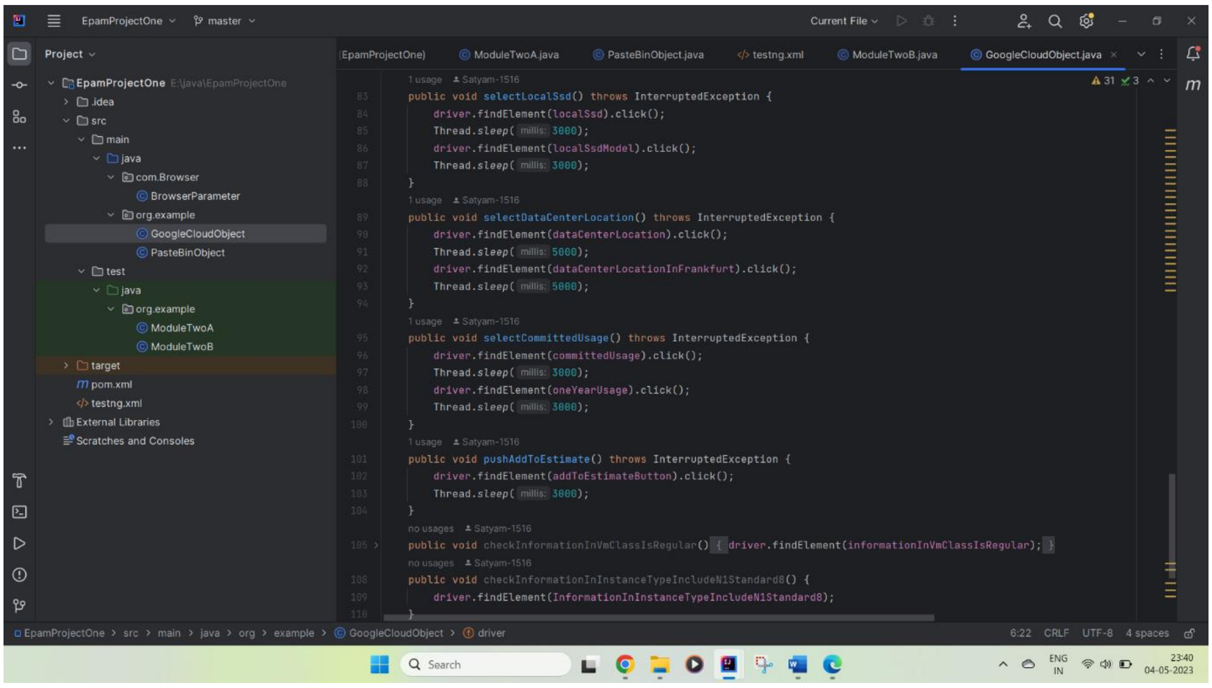
```
1 usage
private By gpuTypeModel =By.xpath( xpathExpression: "//md-option[@value='NVIDIA_TESLA_V100']");
1 usage
private By numberOfGpus = By.xpath( xpathExpression: "//md-select[@placeholder='Number of GPUs']");
1 usage
private By numberOfGpusModel = By.cssSelector("md-option[value='1'][class='ng-scope md-ink-ripple'][ng-disabled]");
1 usage
private By localSsd = By.xpath( xpathExpression: "//md-select[@placeholder='Local SSD']");
1 usage
private By localSsdModel = By.xpath( xpathExpression: "//md-option[@ng-repeat='item in listingCtrl.dynamicSsd.computeServer'
1 usage
private By dataCenterLocation = By.xpath( xpathExpression: "//md-select[@placeholder='Datacenter location']");
1 usage
private By dataCenterLocationInFrankfurt = By.xpath( xpathExpression: "//md-option[@value='europe-west3' and contains(@ng-repea
1 usage
private By committedUsage = By.xpath( xpathExpression: "//md-select[@placeholder='Committed usage']");
1 usage
private By oneYearUsage = By.cssSelector("div[class='md-select-menu-container md-active md-clickable']" + " md-option[valu
1 usage
private By addToEstimateButton = By.xpath( xpathExpression: "//form[@name='ComputeEngineForm']//button[contains(text(),'Add to
1 usage
private By informationInVmClassIsRegular = By.xpath( xpathExpression: "//md-select[@aria-label='VM Class: Regular']");
1 usage
private By InformationInInstanceTypeIncludeN1Standard8 = By.xpath( xpathExpression: "//div[contains (text(),'Instance type: n1-
1 usage
private By regionIsFrankfurt = By.xpath( xpathExpression: "//div[contains (text(),'Region: Frankfurt')]");
1 usage
private By localSsdSpace2x375Gib = By.xpath( xpathExpression: "//div[@class='md-text'][text()='24x375 GB']");
1 usage
private By commitmentTermOneYear = By.xpath( xpathExpression: "//div[contains (text(),'Commitment term: 1 Year')]");

1 usage  Satyam-1516
```



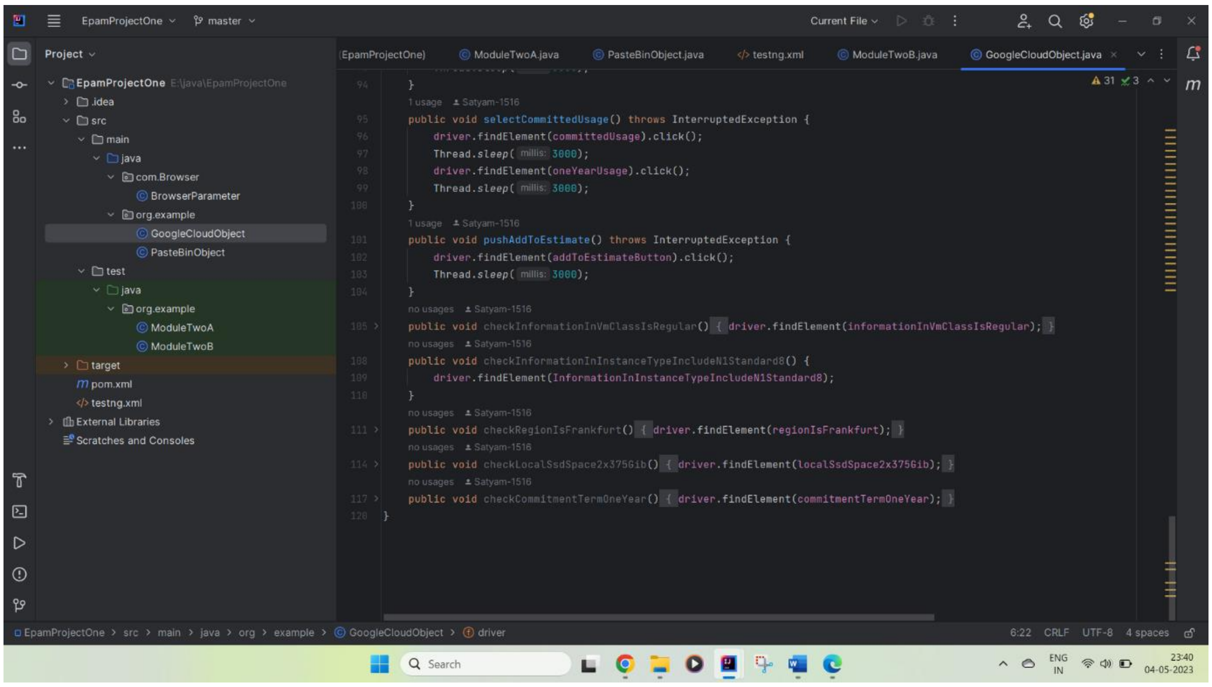
```

1 usage  Satyam-1516
public void homePage() throws InterruptedException {
    WebElement r = driver.findElement(search);
    r.sendKeys( ...keysToSend: "Google Cloud Platform Pricing Calculator");
    // press ENTER
    r.sendKeys(Keys.RETURN);
    Thread.sleep( millis: 5000);
    driver.findElement(searchResult).click();
    Thread.sleep( millis: 10000);
}
1 usage  Satyam-1516
public void sendKeyInToNumberOfInstancesField(String keyForNumberOfInstances) throws InterruptedException {
    WebElement element = driver.findElement(newFirstFrame);
    driver.switchTo().frame(element);
    driver.switchTo().frame( nameOrId: "myFrame");
    Thread.sleep( millis: 3000);
    WebElement numberOfInstancesField = driver.findElement(instancesField);
    numberOfInstancesField.sendKeys(keyForNumberOfInstances);
    Thread.sleep( millis: 3000);
}
1 usage  Satyam-1516
public void selectSeriesOfMachine() throws InterruptedException {
    driver.findElement(seriesOfMachine).click();
    Thread.sleep( millis: 3000);
    driver.findElement(seriesOfMachineModel).click();
    Thread.sleep( millis: 3000);
}
1 usage  Satyam-1516
public void selectMachineType() throws InterruptedException {
    driver.findElement(machineType).click();
    Thread.sleep( millis: 3000);
```



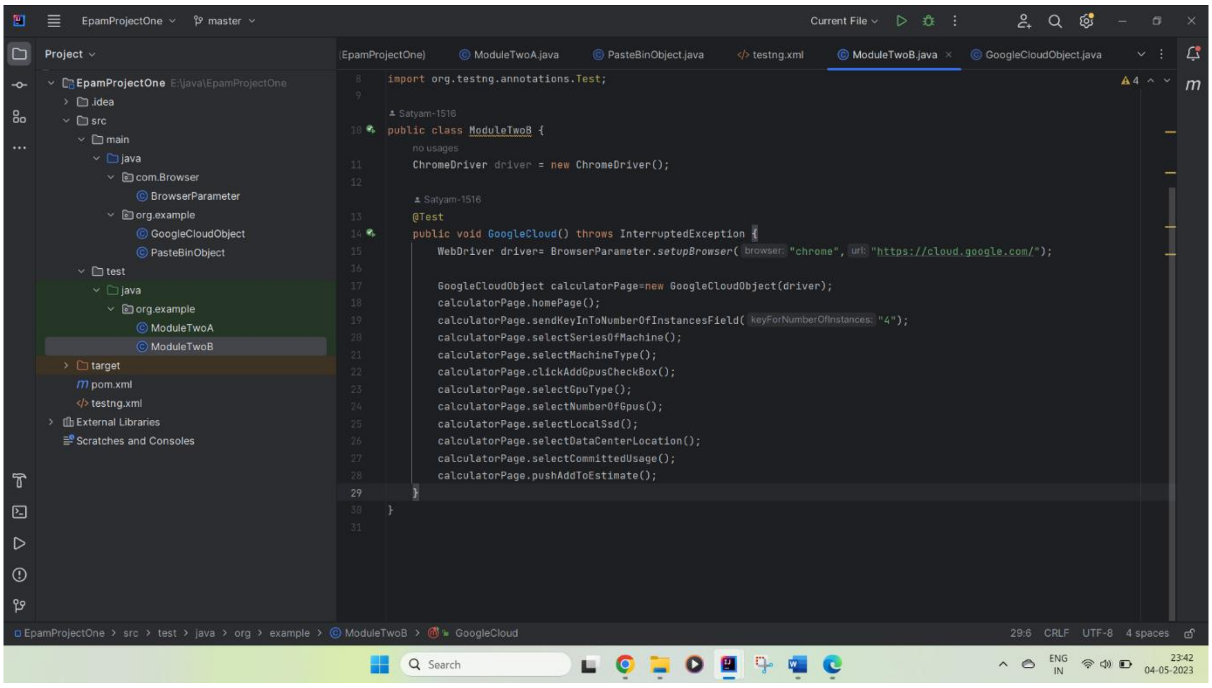
```java
    1 usage  Satyam-1516
    public void selectMachineType() throws InterruptedException {
        driver.findElement(machineType).click();
        Thread.sleep( millis: 3000);
        driver.findElement(computeEngine).click();
        Thread.sleep( millis: 3000);
    }
    1 usage  Satyam-1516
    public void clickAddGpusCheckBox() throws InterruptedException {
        driver.findElement(gpusCheckBox).click();
        Thread.sleep( millis: 3000);
    }
    1 usage  Satyam-1516
    public void selectGpuType() throws InterruptedException {
        driver.findElement(gpuType).click();
        Thread.sleep( millis: 3000);
        driver.findElement(gpuTypeModel).click();
        Thread.sleep( millis: 2000);
    }
    1 usage  Satyam-1516
    public void selectNumberOfGpus() throws InterruptedException {
        driver.findElement(numberOfGpus).click();
        Thread.sleep( millis: 3000);
        driver.findElement(numberOfGpusModel).click();
        Thread.sleep( millis: 3000);
    }
    1 usage  Satyam-1516
    public void selectLocalSsd() throws InterruptedException {
        driver.findElement(localSsd).click();
        Thread.sleep( millis: 3000);
        driver.findElement(localSsdModel).click();
        Thread.sleep( millis: 3000);
```



```java
    1 usage  Satyam-1516
    public void selectLocalSsd() throws InterruptedException {
        driver.findElement(localSsd).click();
        Thread.sleep( millis: 3000);
        driver.findElement(localSsdModel).click();
        Thread.sleep( millis: 3000);
    }
    1 usage  Satyam-1516
    public void selectDataCenterLocation() throws InterruptedException {
        driver.findElement(dataCenterLocation).click();
        Thread.sleep( millis: 5000);
        driver.findElement(dataCenterLocationInFrankfurt).click();
        Thread.sleep( millis: 5000);
    }
    1 usage  Satyam-1516
    public void selectCommittedUsage() throws InterruptedException {
        driver.findElement(committedUsage).click();
        Thread.sleep( millis: 3000);
        driver.findElement(oneYearUsage).click();
        Thread.sleep( millis: 3000);
    }
    1 usage  Satyam-1516
    public void pushAddToEstimate() throws InterruptedException {
        driver.findElement(addToEstimateButton).click();
        Thread.sleep( millis: 3000);
    }
    no usages  Satyam-1516
    public void checkInformationInVmClassIsRegular() { driver.findElement(informationInVmClassIsRegular); }
    no usages  Satyam-1516
    public void checkInformationInInstanceTypeIncludeN1Standard8() {
        driver.findElement(InformationInInstanceTypeIncludeN1Standard8);
    }
```

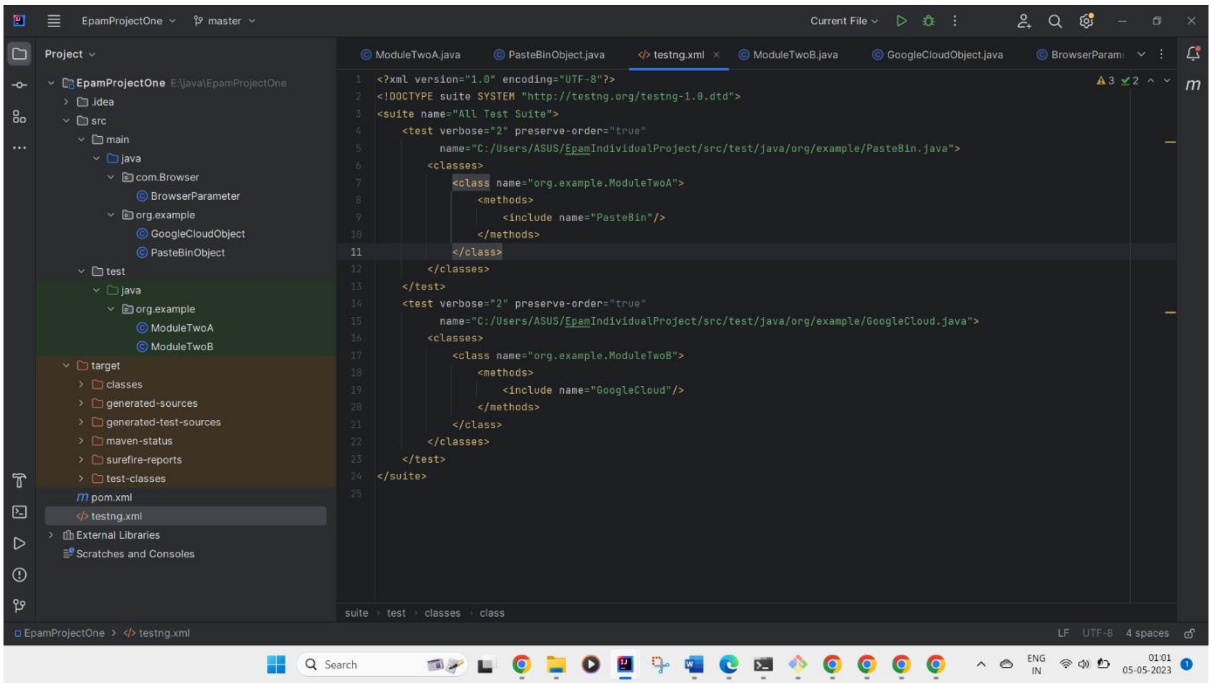


```java
    }
    public void selectCommittedUsage() throws InterruptedException {
        driver.findElement(committedUsage).click();
        Thread.sleep(3000);
        driver.findElement(oneYearUsage).click();
        Thread.sleep(3000);
    }
    public void pushAddToEstimate() throws InterruptedException {
        driver.findElement(addToEstimateButton).click();
        Thread.sleep(3000);
    }
    public void checkInformationInVmClassIsRegular() { driver.findElement(informationInVmClassIsRegular); }
    public void checkInformationInInstanceTypeIncludeN1Standard8() {
        driver.findElement(InformationInInstanceTypeIncludeN1Standard8);
    }
    public void checkRegionIsFrankfurt() { driver.findElement(regionIsFrankfurt); }
    public void checkLocalSsdSpace2x375Gib() { driver.findElement(localSsdSpace2x375Gib); }
    public void checkCommitmentTermOneYear() { driver.findElement(commitmentTermOneYear); }
}
```



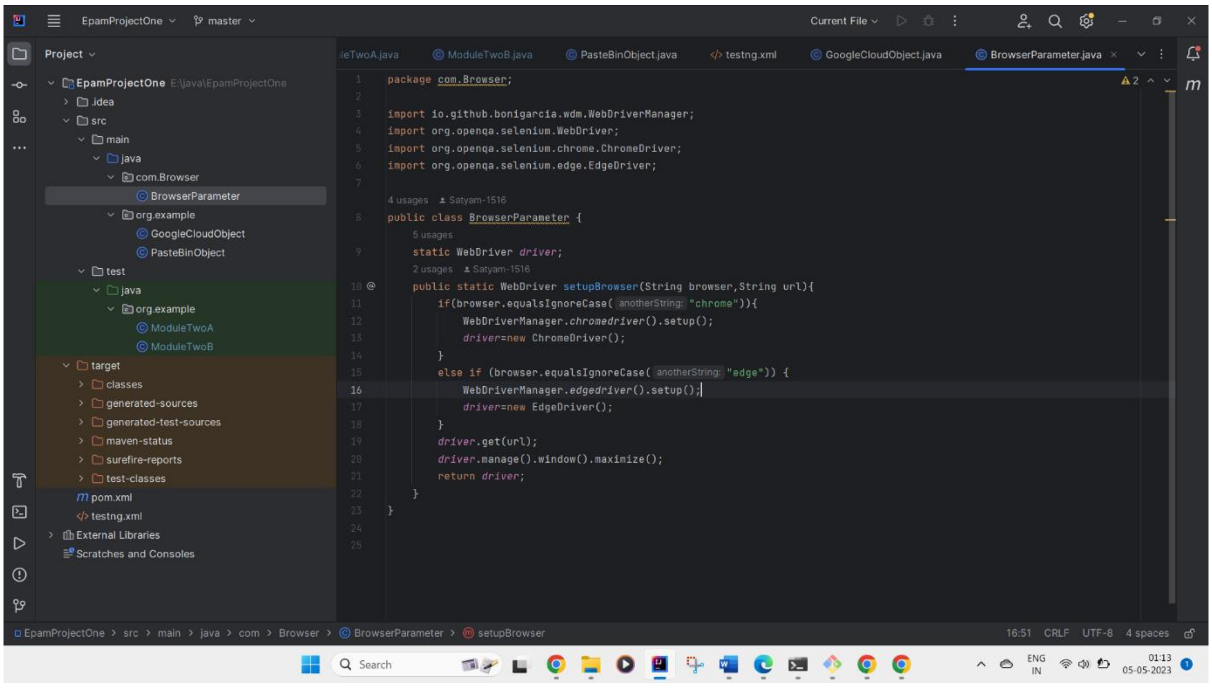
```java
import org.testng.annotations.Test;

public class ModuleTwoB {
    ChromeDriver driver = new ChromeDriver();

    @Test
    public void GoogleCloud() throws InterruptedException {
        WebDriver driver= BrowserParameter.setupBrowser("chrome", "https://cloud.google.com/");

        GoogleCloudObject calculatorPage=new GoogleCloudObject(driver);
        calculatorPage.homePage();
        calculatorPage.sendKeyInToNumberOfInstancesField("4");
        calculatorPage.selectSeriesOfMachine();
        calculatorPage.selectMachineType();
        calculatorPage.clickAddGpusCheckBox();
        calculatorPage.selectGpuType();
        calculatorPage.selectNumberOfGpus();
        calculatorPage.selectLocalSsd();
        calculatorPage.selectDataCenterLocation();
        calculatorPage.selectCommittedUsage();
        calculatorPage.pushAddToEstimate();
    }
}

```

Testng.xml

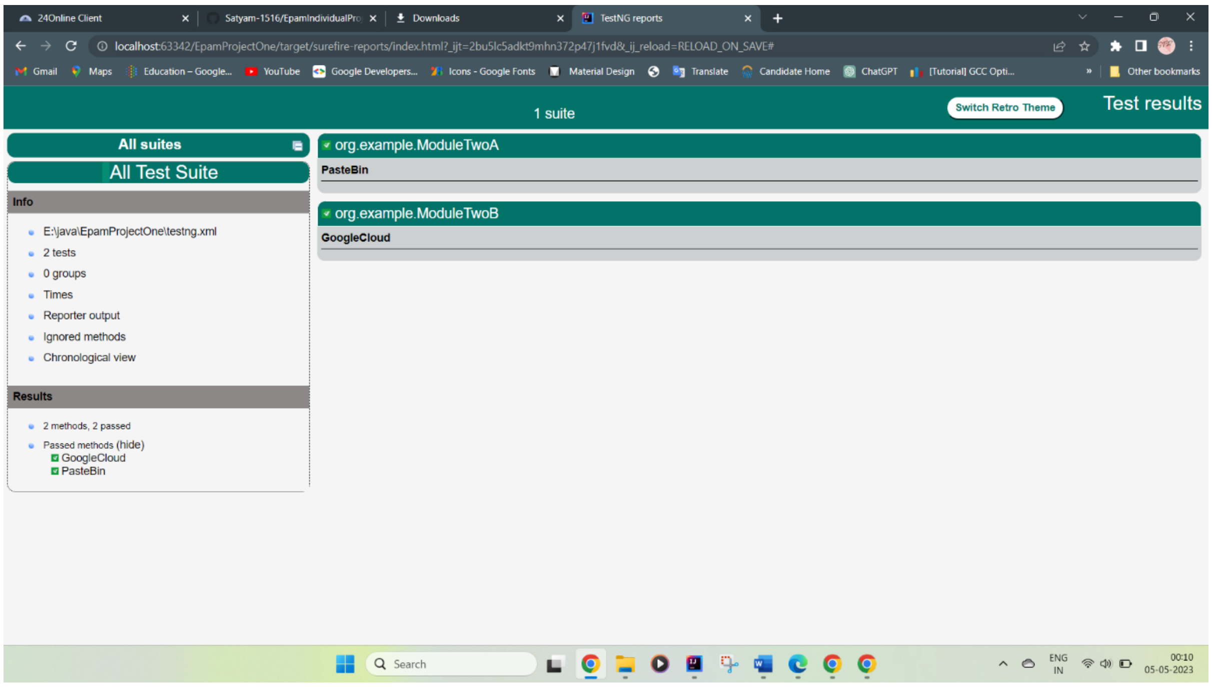


```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="All Test Suite">
    <test verbose="2" preserve-order="true"
          name="C:/Users/ASUS/EpamIndividualProject/src/test/java/org/example/PasteBin.java">
        <classes>
            <class name="org.example.ModuleTwoA">
                <methods>
                    <include name="PasteBin"/>
                </methods>
            </class>
        </classes>
    </test>
    <test verbose="2" preserve-order="true"
          name="C:/Users/ASUS/EpamIndividualProject/src/test/java/org/example/GoogleCloud.java">
        <classes>
            <class name="org.example.ModuleTwoB">
                <methods>
                    <include name="GoogleCloud"/>
                </methods>
            </class>
        </classes>
    </test>
</suite>
```

Browser Parameters



```java
package com.Browser;

import io.github.bonigarcia.wdm.WebDriverManager;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.edge.EdgeDriver;

public class BrowserParameter {
    static WebDriver driver;
    public static WebDriver setupBrowser(String browser,String url){
        if(browser.equalsIgnoreCase("chrome")){
            WebDriverManager.chromedriver().setup();
            driver=new ChromeDriver();
        }
        else if (browser.equalsIgnoreCase("edge")) {
            WebDriverManager.edgedriver().setup();
            driver=new EdgeDriver();
        }
        driver.get(url);
        driver.manage().window().maximize();
        return driver;
    }
}
```


localhost:63342/EpamProjectOne/target/surefire-reports/index.html?_ijt=2bu5lc5adkt9mhn372p47j1fvd&_ij_reload=RELOAD_ON_SAVE#
Switch Retro Theme

1 suite

Test results

All suites

All Test Suite

Info

- E:\java\EpamProjectOne\testng.xml
- 2 tests
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 2 methods, 2 passed
- Passed methods (hide)
  - GoogleCloud
  - PasteBin

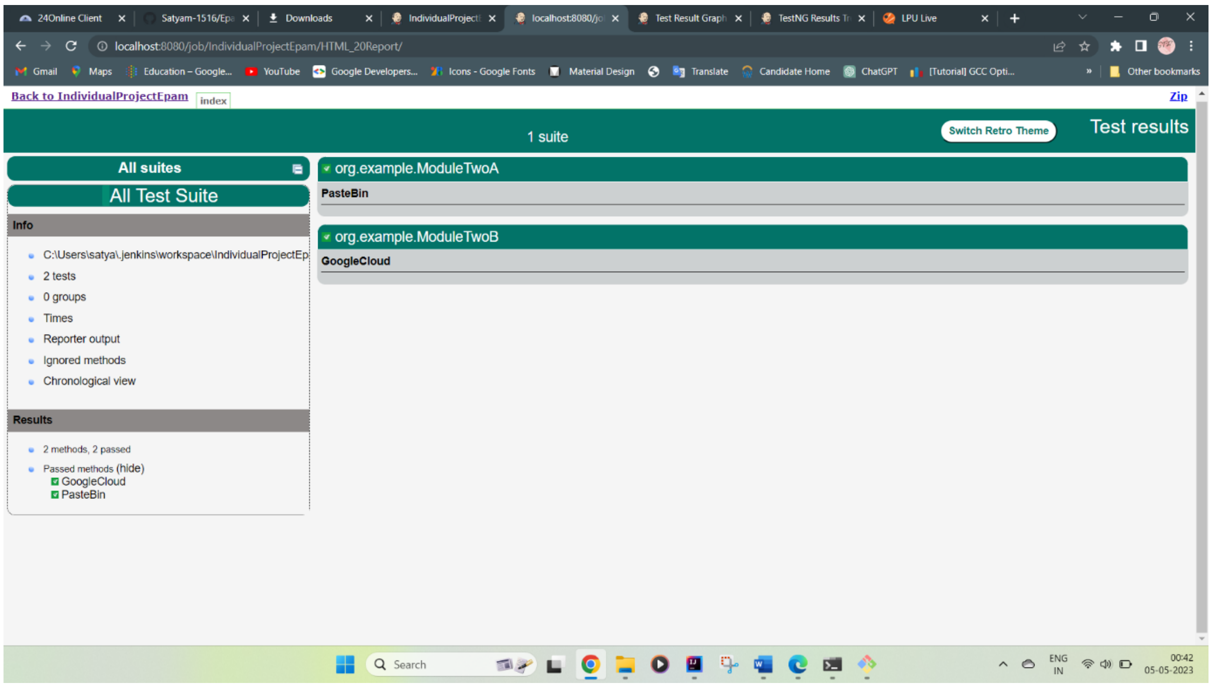
org.example.ModuleTwoA

PasteBin

org.example.ModuleTwoB

GoogleCloud

Tested the above project using Jenkins.


localhost:8080/job/IndividualProjectEpam/HTML_20Report/
Back to IndividualProjectEpam
index
Zip
Switch Retro Theme

1 suite

Test results

All suites

All Test Suite

Info

- C:\Users\satya\.jenkins\workspace\IndividualProjectEp
- 2 tests
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

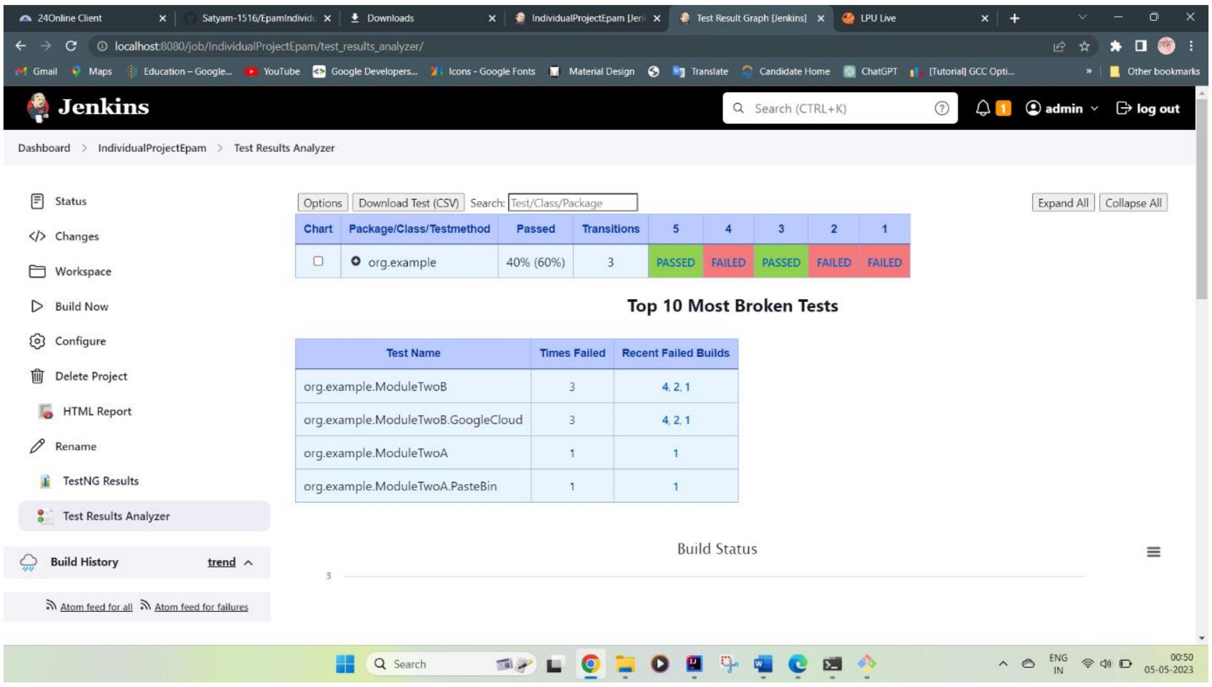
- 2 methods, 2 passed
- Passed methods (hide)
  - GoogleCloud
  - PasteBin

org.example.ModuleTwoA

PasteBin

org.example.ModuleTwoB

GoogleCloud



localhost:8080/job/IndividualProjectEpam/test_results_analyzer/

Jenkins

Dashboard > IndividualProjectEpam > Test Results Analyzer

- Status
- Changes
- Workspace
- Build Now
- Configure
- Delete Project
- HTML Report
- Rename
- TestNG Results
- Test Results Analyzer

Build History trend

Atom feed for all Atom feed for failures

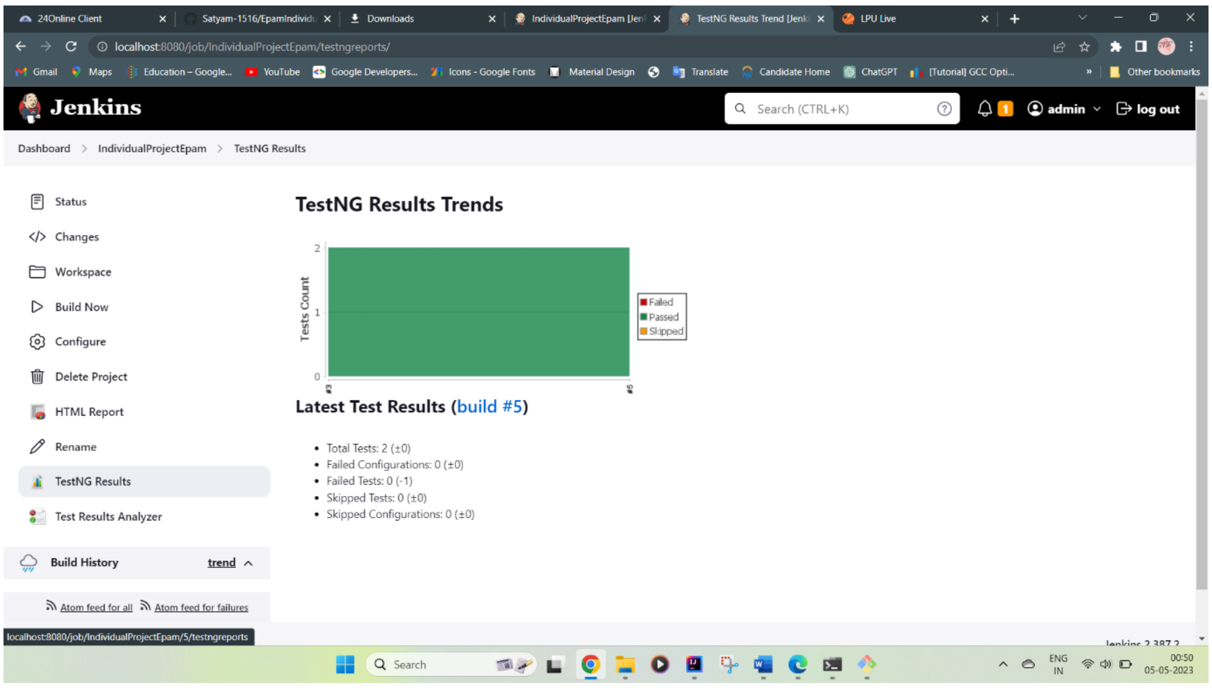
Options | Download Test (CSV) | Search: Test/Class/Package | Expand All | Collapse All

Chart	Package/Class/Testmethod	Passed	Transitions	5	4	3	2	1
☐	org.example	40% (60%)	3	PASSED	FAILED	PASSED	FAILED	FAILED

Top 10 Most Broken Tests

Test Name	Times Failed	Recent Failed Builds
org.example.ModuleTwoB	3	4, 2, 1
org.example.ModuleTwoB.GoogleCloud	3	4, 2, 1
org.example.ModuleTwoA	1	1
org.example.ModuleTwoA.PasteBin	1	1

Build Status



localhost:8080/job/IndividualProjectEpam/testngreports/

Jenkins

Dashboard > IndividualProjectEpam > TestNG Results

- Status
- Changes
- Workspace
- Build Now
- Configure
- Delete Project
- HTML Report
- Rename
- TestNG Results
- Test Results Analyzer

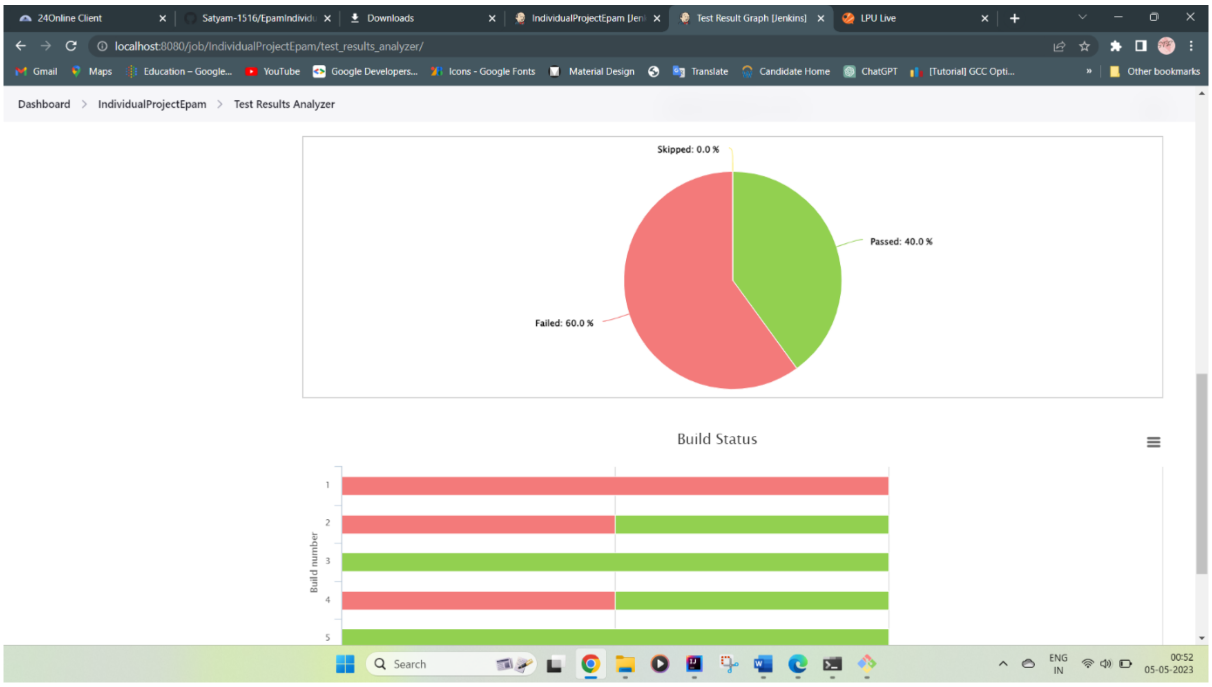
Build History trend

Atom feed for all Atom feed for failures

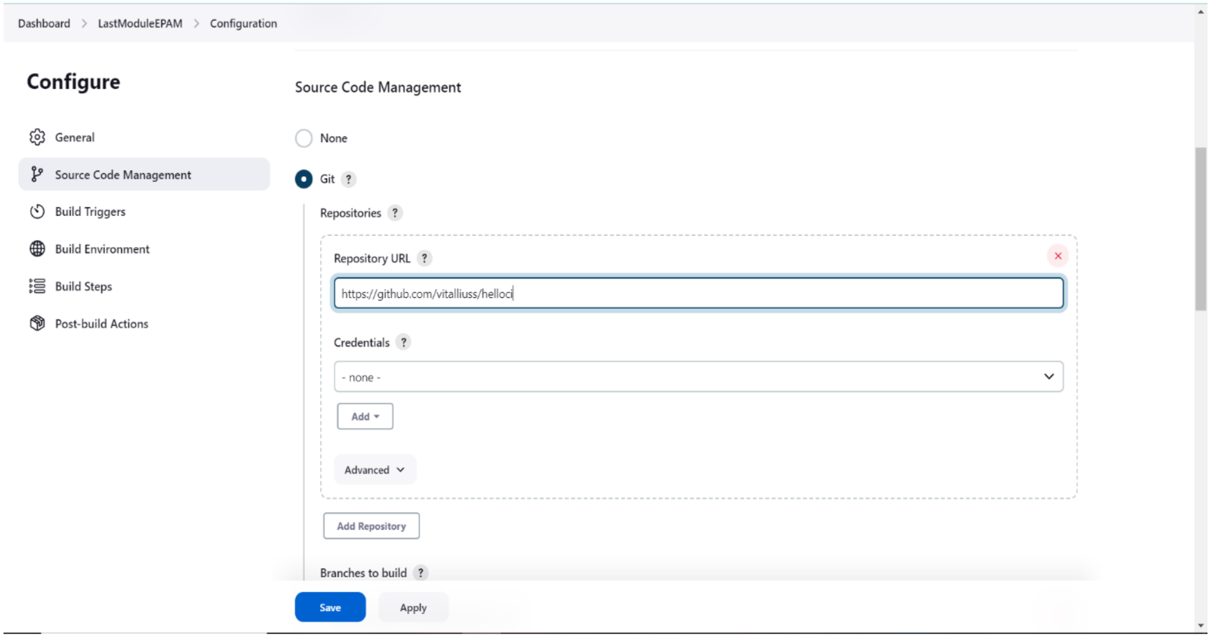
TestNG Results Trends

Latest Test Results (build #5)

- Total Tests: 2 (±0)
- Failed Configurations: 0 (±0)
- Failed Tests: 0 (-1)
- Skipped Tests: 0 (±0)
- Skipped Configurations: 0 (±0)



Clone the project (https://github.com/vitalliuss/helloci) – and set up build triggers so that the task is performed every 5 minutes.



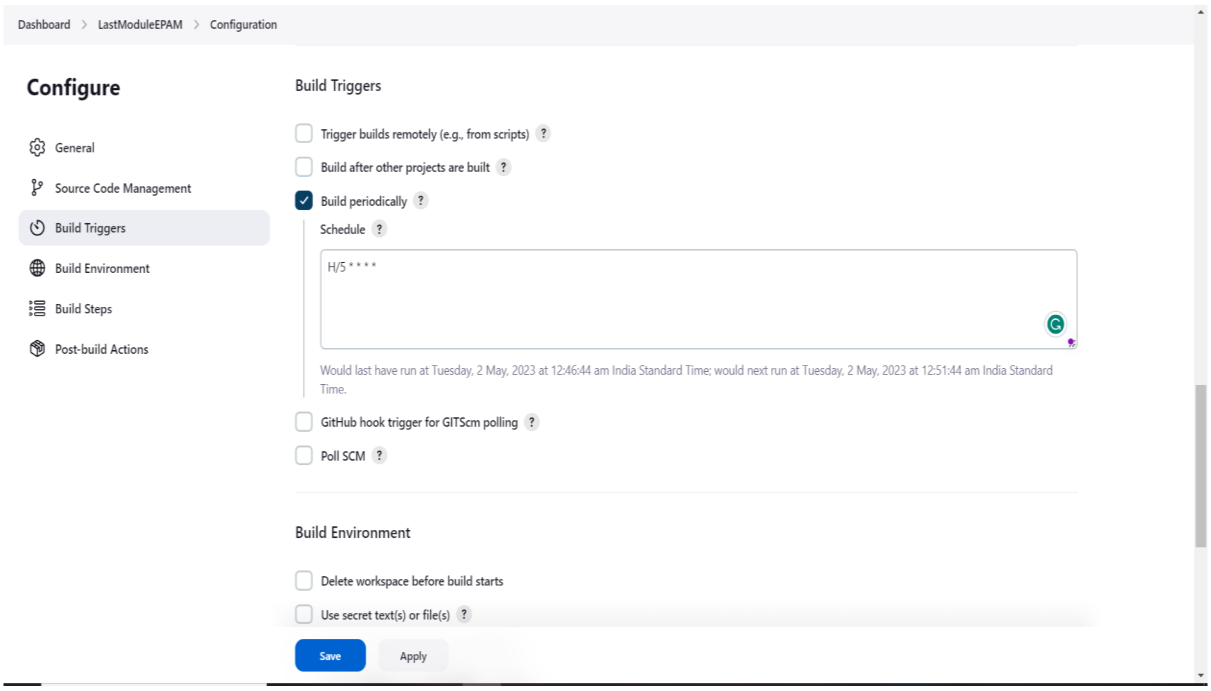
Source Code Management

None [✓ Git]

Repositories

Repository URL: https://github.com/vitalliuss/helloci

Credentials: - none -



Dashboard > LastModuleEPAM > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Build Triggers

- [ ] Trigger builds remotely (e.g., from scripts) ?
- [ ] Build after other projects are built ?
- [x] Build periodically ?

Schedule ?

```
H/5 * * * *
```

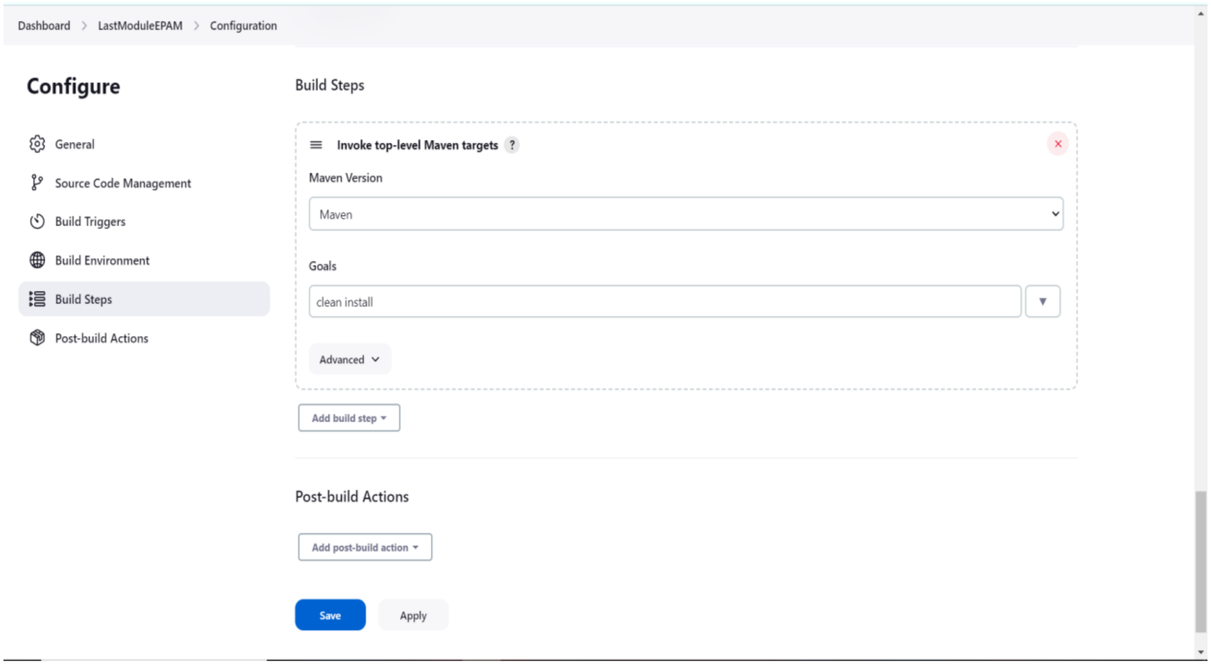
Would last have run at Tuesday, 2 May, 2023 at 12:46:44 am India Standard Time; would next run at Tuesday, 2 May, 2023 at 12:51:44 am India Standard Time.

- [ ] GitHub hook trigger for GITScm polling ?
- [ ] Poll SCM ?

Build Environment

- [ ] Delete workspace before build starts
- [ ] Use secret text(s) or file(s) ?

Save Apply



Dashboard > LastModuleEPAM > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Build Steps

Invoke top-level Maven targets ?

Maven Version: Maven

Goals: clean install

Advanced

Add build step

Post-build Actions

Add post-build action

Save Apply

CHAPTER 4
CONCLUSION

The completion of the Maven practical task, as well as the subsequent tasks, showcases a strong grasp of the Maven build tool and the ability to use it efficiently. The task involved downloading a test project and collecting it with Maven, changing the junit version, and ensuring that the new library version was added to the repository. These tasks demonstrate proficiency in managing dependencies and building projects with Maven.

The Webdriver module's practical tasks demonstrate a solid understanding of Selenium WebDriver, framework unit tests, and Page Object concepts. The "I can win" task involved automating the creation of a new paste on a service like Pastebin with specific attributes, while the "Hurt Me Plenty" task automated the use of the Google Cloud Platform Pricing Calculator with specific parameters. Both tasks demonstrate a solid understanding of the Page Object model and the ability to create efficient, maintainable code.

The successful completion of the Framework practical task involves the development of a robust automation framework for the "Hurt Me Plenty" task. The framework includes a WebDriver manager for managing browser connections, page abstractions using Page Object/Page Factory, models for business objects of the required elements, and property files with test data for at least two different environments. The framework also includes XML suites for smoke tests and other tests, screenshot capture when a test fails, and the ability to run with Jenkins, parameterize browsers, test suites, and environments.

The Amazon UI test involved testing various functionalities of the Amazon web application, including login, wishlist creation, searching and adding products, deleting products, renaming wishlist and deleting wishlist. The test demonstrates proficiency in using Selenium WebDriver, TestNG, and Page Object pattern to automate various tasks, ensuring that the application functions as expected. The successful completion of the test shows that the test cases were comprehensive, and the application meets the necessary requirements.

Overall, the completion of these tasks and the successful execution of the Amazon UI test demonstrate a strong understanding of various technologies such as Maven, Selenium WebDriver, TestNG, Jenkins, and Page Object pattern. These skills are essential for developing and maintaining robust automation frameworks, ensuring that web applications meet the necessary requirements and function as expected.