Testing Automation using JAVA and Selenium

EPAM
A Training Report

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

Bachelor of Technology
Computer Science and Engineering
(Web Development)

LOVELY PROFESSIONAL UNIVERSITY
PHAGWARA, PUNJAB

From 15/01/2023 to 28/04/2023

SUBMITTED BY
Name of Student: Naik Ruturaj Pandit
Registration Number: 11905412

SUBMITTED TO
Name of Supervisor: Miss. Sakshi

Student Declaration

To whom so ever it may concern,

I, Naik Ruturaj Pandit, 11905412, hereby declare that the work done by me on **"TESTING AUTOMATION USING JAVA AND SELENIUM"** from 6th Jan-2023 to present, under the internal supervisor – Sakshi, 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 of the Student (Registration Number)

Naik Ruturaj Pandit (11905412)

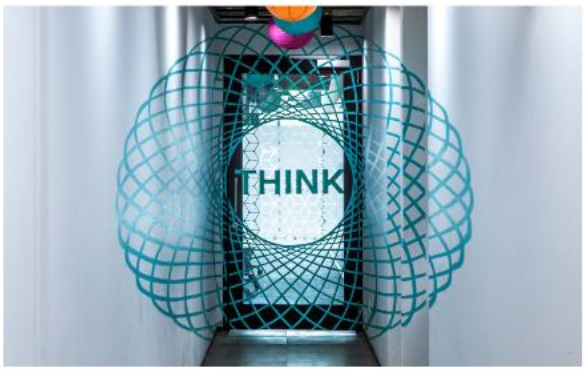
Ruturaj

Signature of the student Dated: 01/05/2023

Table of Contents

CHAPTER 1
INTRODUCTION TO COMPANY

1.1. About EPAM :



EPAM Systems, Inc. is a leading global provider of digital platform engineering and software development services. The company was founded in 1993 and is headquartered in Newtown, Pennsylvania, USA. With over 41,000 professionals in 35+ countries, EPAM works with Fortune 500 companies, emerging start-ups, and leading technology firms to design, develop, and deliver innovative solutions that transform businesses and improve the lives of people around the world.

EPAM's core services include product engineering, digital platform engineering, enterprise application development, digital experience design, and consulting. The company's expertise spans a wide range of industries, including finance, healthcare, retail, media and entertainment, travel and hospitality, and automotive. EPAM is committed to providing its clients with the highest level of quality, agility, and scalability, leveraging its global delivery model and best-in-class practices in software engineering, design thinking, and agile methodologies.

EPAM has been consistently recognized by industry analysts and leading publications for its exceptional performance and innovative solutions. The company has been named a Leader in the Gartner Magic
Quadrant for Application Services, a Leader in the IDC Markets cape for Worldwide Digital Transformation Consulting and Systems Integration Services, and a Top 100 Outsourcing Company by IAOP for several years in a row.
EPAM is also committed to making a positive impact on society through its Corporate Social Responsibility (CSR) initiatives. The company's CSR programs focus on education, innovation, and community development, providing support and resources to students, entrepreneurs, and social enterprises around the world.

EPAM is a global leader in digital platform engineering and software development services, providing innovative solutions to transform businesses and improve lives. The company's commitment to quality, agility, and scalability, combined with its global delivery model and expertise in software engineering, design thinking, and agile methodologies, make it a trusted partner for Fortune 500 companies and emerging start up’s alike.

EPAM Systems, Inc. was founded in 1993 in Minsk, Belarus, as a small software development company. Over the years, the company has expanded significantly and today it has a global presence, with offices and delivery centres in North America, Europe, Asia, and Latin America.

EPAM's core services include product engineering, digital platform engineering, enterprise application development, digital experience design, and consulting. The company's product engineering services focus on developing innovative software products and solutions that meet the evolving needs of its clients. Its digital platform engineering services help clients build and operate scalable, secure, and robust digital platforms that drive business growth and competitive advantage. Its enterprise application development services include developing custom enterprise applications, modernizing legacy systems, and integrating third-party applications. Its digital experience design services help clients create engaging and intuitive digital experiences that enhance customer satisfaction and loyalty. Its consulting services provide clients with strategic advice and guidance on digital transformation, innovation, and business process optimization.

EPAM's clients come from a wide range of industries, including finance, healthcare, retail, media and entertainment, travel and hospitality, and automotive. Some of its notable clients include Capital One, UBS, Coca-Cola, Siemens, and Google.

EPAM has a strong focus on innovation and has invested heavily in building its digital platform engineering capabilities, including expertise in cloud computing, big data analytics, artificial intelligence, and machine learning. The company has also established strategic partnerships with leading technology vendors such as Amazon Web Services, Microsoft, and Salesforce to provide its clients with the best-in-class technology solutions.
EPAM is committed to quality and has established a rigorous quality management system that ensures the delivery of high-quality solutions to its clients. The company has also adopted best-in-class practices in software engineering, design thinking, and agile methodologies to ensure the agility, scalability, and adaptability of its solutions.

EPAM has received numerous awards and recognitions for its exceptional performance and innovative solutions. For example, the company was named a Leader in the Gartner Magic Quadrant for Application Services for several years in a row, a Leader in the IDC Markets cape for Worldwide Digital Transformation Consulting and Systems Integration Services, and a Top 100 Outsourcing Company by IAOP.

EPAM is a leading global provider of digital platform engineering and software development services, with a strong focus on innovation, quality, and customer satisfaction. Its extensive expertise, global delivery model, and best-in-class practices make it a trusted partner for businesses looking to transform and thrive in the digital age.

Company's Mission Statement:

"Our teams of technologists, strategists and designers transform our customers' business through a combination of engineering expertise, design thinking and business consulting."

Since 1993, EPAM Systems, Inc. (NYSE: EPAM) has leveraged its software engineering expertise to become a leading global product development, digital platform engineering, and top digital and product design agency. Through its 'Engineering DNA' and innovative strategy, consulting, and design capabilities, EPAM works in collaboration with its customers to deliver next-gen solutions that turn complex business challenges into real business outcomes. EPAM's global teams serve customers in more than 35 countries across North America, Europe, Asia and Australia. As a recognized market leader in multiple categories among top global independent research agencies, EPAM was one of only four technology companies to appear on Forbes 25 Fastest Growing Public Tech Companies list every year of publication since 2013 and ranked as the top IT services company on Fortune's 100 Fastest-Growing Companies list in 2019 and 2020.

EPAM Systems Employee Benefits

- **Health & Wellness**

  Dental Benefits, Medical Insurance, Disability Insurance, Life Insurance, Health Savings Account (HSA), Flexible Spending Account (FSA), Vision Benefits

- **Workplace Culture**

  Flexible Work Arrangements, Work from Home, Professional Development, Team Building Events

- **Financial Benefits & Perks**

  Equity Incentive Plan, Performance Bonuses, Discounts, Commuter Benefits, 401k (with matching)

- **Life Outside of Work**

  Back-up Dependent Care Program, Employee Assistance Program, Paid Parental Leave, Paid Holidays, Paid Vacations

ORIGIN AND GROWTH OF COMPANY

EPAM Systems is a global software engineering and IT consulting company that was founded in 1993 by Arkadiy Dobkin, a Belarusian-American entrepreneur. The company's name is an acronym of its original name, "Effective Programming for America."

EPAM started as a small software development company in Minsk, Belarus, with just four employees. In the early years, the company provided software development services to clients in the United States and Europe, mainly in the financial services industry.

In 2002, EPAM opened its first office in the United States, in Princeton, New Jersey. This marked the company's expansion into the North American market. In subsequent years, EPAM continued to grow and expand its services, opening offices in many other countries, including Canada, Mexico, Argentina, Brazil, China, India, Poland, Russia, and the United Kingdom.

Today, EPAM has become one of the world's leading software engineering and IT consulting firms, with more than 47,000 employees in over 35 countries. The company serves clients in various industries, including financial services, healthcare, media and entertainment, retail and distribution, and software and technology. EPAM is also recognized as a leader in digital platform engineering, digital strategy consulting, and customer experience design.

Various departments and their functions of EPAM

EPAM Systems is a global software engineering services company that provides software development, digital platform engineering, and product design services. The company operates through various departments, each with its unique functions. Here are some of the key departments and their functions at EPAM:

1. Delivery: This department is responsible for delivering software development and engineering services to clients. The delivery team manages project timelines, resources, and quality control.

2. Engineering: The engineering department is responsible for developing new software applications and tools, improving existing ones, and managing the technology infrastructure.

3. Design: This department is responsible for creating user interfaces, user experience design, and visual design for software applications and digital platforms.

4. Quality Assurance: The QA department is responsible for ensuring that software applications and platforms meet quality standards, including functionality, performance, and security.

5. Business Analysis: The business analysis department is responsible for understanding client needs and defining software requirements. They work closely with the delivery and engineering teams to ensure that the software developed meets client needs.

6. Sales and Marketing: This department is responsible for promoting EPAM's services and building relationships with clients.

7. Human Resources: The human resources department is responsible for recruiting and managing EPAM's employees, ensuring that they have the necessary skills and training to deliver high-quality services to clients.

8. Finance: The finance department is responsible for managing EPAM's finances, including budgeting, forecasting, and financial reporting. Overall, these departments work together to provide comprehensive software development and engineering services to clients worldwide.

CHAPTER 2 : TECHNOLOGIES LEARNT DURING INTERNSHIP

Throughout the period of the internship, we learned a lot of skills. Every skill that we learned or gained have different scopes and each skill plays a vital role in our lives. The scope of some skills that we learned are mentioned below:

- Teamwork
- Problem solving
- Co-ordination
- Interpersonal Skills
- Time Management

ABOUT THE SKILLS AND TECHNOLOGIES I LEARNT :

TESTING

Testing is a crucial process in software development that aims to ensure that the software being developed meets the desired quality and requirements. It is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is essential to ensure that software systems are reliable, robust, secure, and perform well under various conditions.

Software testing involves a variety of techniques and methodologies that are used to identify defects or errors in the software.
There are two types of testing, one is ma nual testing which is performed manually and another is automation testing.

Manual testing is a process of software testing where a person, commonly known as a tester, manually executes test cases without using any automated testing tools. It is the oldest form of testing, and despite the advent of automation, manual testing remains an essential part of software testing.

Manual testing is typically done in a step-by-step approach to ensure that the software application is free from errors and functions as per the specified requirements. The manual testing process can be broadly divided into the following stages:

- Test Planning: Test planning is the first stage of manual testing. In this stage, the tester defines the objectives of the test, identifies the scope of testing, and prepares a test plan that outlines the test strategy, test cases, and test schedule.

- Test Design: In this stage, the tester designs test cases that define the input data, expected output, and the steps to be followed to execute the test case. The test cases are designed based on the functional and non-functional requirements of the software application.

- Test Execution: This is the stage where the actual testing takes place. The tester executes the test cases, records the results, and reports any defects found during the testing process. The tester also communicates with the development team to resolve any issues found during the testing process.

- Test Reporting: The test reporting stage involves creating test reports that summarize the test results. The test reports include information about the test cases executed, defects found, and the status of the testing process.

Manual testing can be performed in various forms, such as functional testing, regression testing, system testing, integration testing, and user acceptance testing. The testers may also use various manual testing techniques such as exploratory testing, ad-hoc testing, and monkey testing to identify any potential issues with the software application.

Automation testing refers to the use of software tools to execute and verify tests automatically, without the need for human intervention. It is a valuable technique for software testing, as it helps to speed up the testing process, reduce the risk of errors and enhance the accuracy and consistency of test results.

There are various types of automation testing techniques, such as functional testing, regression testing, performance testing, and load testing. Each of these techniques has its specific benefits and purposes.

Functional testing is used to verify the software's functionality and ensures that it meets the requirements and specifications. Regression testing is used to ensure that changes or updates in the software do not affect its existing functionality. Performance testing is used to test the software's performance under varying workloads, while load testing is used to test the software's ability to handle the expected user load.

Automation testing tools are essential for carrying out automated tests. These tools come with a wide range of features and capabilities that allow testers to create, execute, and manage automated test cases effectively. Some popular automation testing tools include Selenium, Appium, TestComplete, and HP UFT.

One of the significant benefits of automation testing is that it enables testers to perform repetitive tests efficiently and quickly. Testers can also test a large number of scenarios and use cases with automation testing, which would be challenging to achieve manually.

Automation testing also provides a high level of accuracy and consistency in test results. Since the tests are automated, the possibility of human error is reduced, and the test results are consistent every time the tests are run.

Another advantage of automation testing is that it reduces the time and cost involved in testing. Once the tests are automated, they can be run repeatedly without the need for additional resources. This saves time and reduces the overall cost of testing.

In conclusion, automation testing is an essential technique for software testing, providing many benefits over manual testing. It helps to speed up the testing process, reduce the risk of errors, and enhance the accuracy and consistency of test results. With the increasing complexity of software systems, automation testing has become a crucial component of software development and testing.

JAVA

Java is a high-level programming language that is widely used for developing software applications. It was developed by James Gosling at Sun Microsystems (now a part of Oracle Corporation) in the mid-1990s, with the aim of creating a platform-independent language that could run on any hardware or operating system. Java's popularity and versatility have made it a go-to language for developers across a range of industries.

One of the key features of Java is its "write once, run anywhere" (WORA) philosophy. This means that a Java program can be written on one platform and run on any other platform without any modifications. Java achieves this through its Java Virtual Machine (JVM), which acts as an interpreter between the Java code and the underlying hardware and operating system.

Java is also an object-oriented language, which means that programs are built using objects that encapsulate data and 14ehaviour. This makes it easier to organize and manage complex code, and allows developers to reuse code across different projects. Java also has a rich set of libraries and frameworks, which make it easy to build complex applications with minimal effort.

Some of the key applications of Java include web applications, desktop applications, mobile applications, enterprise applications, and games. Java is particularly well-suited for large-scale enterprise applications, as it can handle multiple threads and has built-in support for network protocols like TCP/IP.

In terms of syntax, Java is similar to other C-based languages like C++ and C#. It has a simple, easy-to-learn syntax that is designed to be easy to read and write. Some of the key features of Java include strong typing, garbage collection, and automatic memory management.

Overall, Java is a versatile and powerful programming language that has stood the test of time. It continues to be widely used across a range of industries, and is an excellent choice for developers who are looking to build complex, enterprise-level applications.

SELENIUM

Selenium is an open-source web testing tool that is commonly used for automating web browsers. It allows users to write scripts in a variety of programming languages, including Java, Python, and C#, to control web browsers and perform automated testing on web applications.

Some key features of Selenium include:

- Cross-browser compatibility: Selenium supports multiple web browsers such as Chrome, Firefox, Safari, and Internet Explorer.
- Flexibility: Selenium can be used with a variety of programming languages and integrated development environments (IDEs).
- Testing capabilities: Selenium supports a wide range of testing types, including functional testing, regression testing, and load testing.
- Extensibility: Selenium can be extended with plugins and custom code to meet specific testing needs.

There are several components to Selenium, including:

- Selenium IDE: A record-and-playback tool for creating and executing Selenium scripts in a browser.
- Selenium WebDriver: A library that allows users to control web browsers programmatically through code.
- Selenium Grid: A tool for distributing Selenium tests across multiple machines and browsers.
- Selenium is widely used in the software testing industry, particularly for testing web applications. It can help to automate repetitive tasks, reduce testing time, and improve the overall quality of software applications.

Selenium was first developed by Jason Huggins in 2004 as an internal tool at ThoughtWorks, a software development and consulting company.
Selenium is open-source software and is released under the Apache 2.0 license.

Selenium WebDriver is the most commonly used component of Selenium and allows users to interact with web browsers in a more programmatic way compared to Selenium IDE.
Selenium is compatible with various operating systems, including Windows, macOS, and Linux.

Selenium supports a wide range of programming languages such as Java, Python, C#, Ruby, JavaScript, and more. Selenium is used by many companies, including Google, Facebook, Amazon, and Microsoft, to automate their web testing processes.

Selenium is not just limited to web testing, it can also be used for web scraping, browser automation, and web-based administration tasks. Selenium has a large community of users and developers who contribute to its development and maintenance through open-source projects and forums. Selenium's popularity has led to the development of several other tools and frameworks, such as Appium for mobile testing and Cypress for modern web applications.

Page Object Model (POM) and **Page Factory Model** are two design patterns that help to improve the organization, readability, and maintainability of automated tests.

Page Object Model (POM) is a design pattern that suggests creating a separate class for each page of the application under test. Each class represents a single page and contains all the elements and actions that can be performed on that page. This approach helps to encapsulate the functionality of the page and reduces code duplication, making the test code more maintainable.

Page Factory is an extension of the POM pattern. It uses the concept of annotations to initialize the web elements in a page class. With Page Factory, you can create an instance of the page class and then use the @FindBy annotation to initialize the web elements. This approach helps to reduce the code for element initialization and improves the readability of the test code.

The Page Object Model (POM) is a design pattern used to improve the maintainability and readability of test automation code. It suggests creating a separate class for each page of the application under test. Each page class contains all the web elements and actions related to that

page. This approach helps to encapsulate the functionality of the page and reduces code duplication.

The Page Factory model is an extension of the POM pattern. It uses the concept of annotations to initialize the web elements in a page class. With Page Factory, you can create an instance of the page class and then use the @FindBy annotation to initialize the web elements. This approach helps to reduce the code for element initialization and improves the readability of the test code.

JIRA SOFTWARE

Jira is a widely used project management software that is designed to help teams plan, track, and manage their work more efficiently. It is primarily used by software development teams, but can also be used by teams in other industries such as marketing, HR, and operations.

Jira software allows teams to create and track issues, bugs, and tasks in a single, centralized location. This makes it easier for team members to stay on top of their work and for managers to track progress and identify areas where improvements can be made.

One of the key features of Jira software is its customizable workflows. This allows teams to create custom processes for how they want to manage their work. For example, a software development team may have a workflow that includes stages such as planning, development, testing, and deployment. Jira software is a powerful and flexible tool that can help teams of all sizes and industries manage their work more effectively. With its customizable workflows, collaboration features, and powerful reporting capabilities, it is a valuable tool for any team looking to improve their project management processes.

JENKINS

Jenkins is a popular open-source automation server that is used for continuous integration and continuous delivery (CI/CD) of software projects. It is a highly extensible platform that can be customized with a wide range of plugins to suit the needs of any development team.

The basic concept behind Jenkins is that it automates the process of building, testing, and deploying code. Whenever a developer commits code to a code repository, Jenkins can be

configured to automatically build the code, run tests, and deploy the code to various environments.

Jenkins provides a user-friendly web interface that makes it easy to create, configure, and manage jobs. Jobs are the basic building blocks of Jenkins, and they define the steps that Jenkins will execute in order to build, test, and deploy code.

Jenkins can be used with a variety of programming languages, build tools, and testing frameworks, including Java, Python, Ruby, Node.js, Maven, Gradle, and many others. This flexibility makes Jenkins a popular choice for teams that use a wide range of tools and technologies.

One of the key features of Jenkins is its plugin architecture. Jenkins has a large and active community of developers who have created hundreds of plugins that extend the functionality of Jenkins. Plugins can be used to integrate Jenkins with other tools and services, such as GitHub, Bitbucket, Docker, and many others.

Jenkins also provides support for distributed builds, which means that it can be used to build and test code across multiple machines or even across multiple geographic locations. This makes it possible for development teams to scale their build and test processes as needed, without having to invest in expensive hardware.

TestNG AND JUNIT

Junit and TestNG are two popular Java testing frameworks used by software developers for unit testing and integration testing.

Junit is the most widely used testing framework for Java applications. It provides a simple and easy-to-use API for writing and running unit tests. Junit supports annotations such as @Test, @Before, @After, @BeforeClass, and @AfterClass to define test methods, setup methods, and teardown methods. Junit also provides a rich set of assertions to verify the expected 18ehaviour of the code under test.

TestNG is another popular testing framework for Java applications that provides additional features beyond Junit. TestNG supports test suites, which allow developers to group test cases into logical units for better organization and execution. TestNG also supports parameterized tests, which enable developers to run the same test case with different input parameters. TestNG also provides more flexible configuration options and supports parallel test execution.

Both Junit and TestNG are powerful and widely used testing frameworks in the Java ecosystem, and the choice of framework often depends on the specific needs and requirements of the project.

Junit has a simple and straightforward API, making it easy to learn and use for writing unit tests. It also has a large and active community, with many resources available online for learning and troubleshooting. However, Junit can be limited in its capabilities when it comes to more complex testing scenarios, such as testing multi-threaded code or testing code that relies on external resources.

TestNG, on the other hand, provides more advanced features and flexibility for testing complex scenarios. TestNG's support for test suites and parameterized tests makes it a popular choice for integration testing and end-to-end testing. TestNG also provides better support for testing multi-threaded code and for running tests in parallel. However, TestNG can be more complex to learn and use than Junit, and may require more configuration and setup.

POSTMAN

Postman is a popular API development tool that simplifies the process of designing, testing, and documenting APIs. It provides developers with a comprehensive interface for creating and managing API requests, responses, and endpoints, making it easier to build and test APIs.

With Postman, developers can create requests using various HTTP methods such as GET, POST, PUT, DELETE, and more. They can also send headers and parameters, set authentication options, and add pre-request scripts to automate the testing process. Additionally, Postman provides several features for testing and debugging APIs, such as the ability to save and share collections of API requests, and a testing framework that allows for the creation of automated tests.

Postman also simplifies the process of documenting APIs, which is an essential step in the development process. With Postman, developers can easily generate API documentation in various formats, including HTML, Markdown, and Swagger.

One of the significant benefits of Postman is its user-friendly interface. The platform is designed to be intuitive and easy to use, even for developers who are not familiar with API development. Additionally, Postman provides extensive documentation and resources to help developers get started and troubleshoot any issues that arise.

Overall, Postman is an essential tool for API development, testing, and documentation. It simplifies the process of building and managing APIs, allowing developers to focus on creating high-quality applications without worrying about the technical details of API development.

CHAPTER 3: INTRODUCTION OF PROJECT UNDERTAKEN

ROLE UNDERTAKEN:

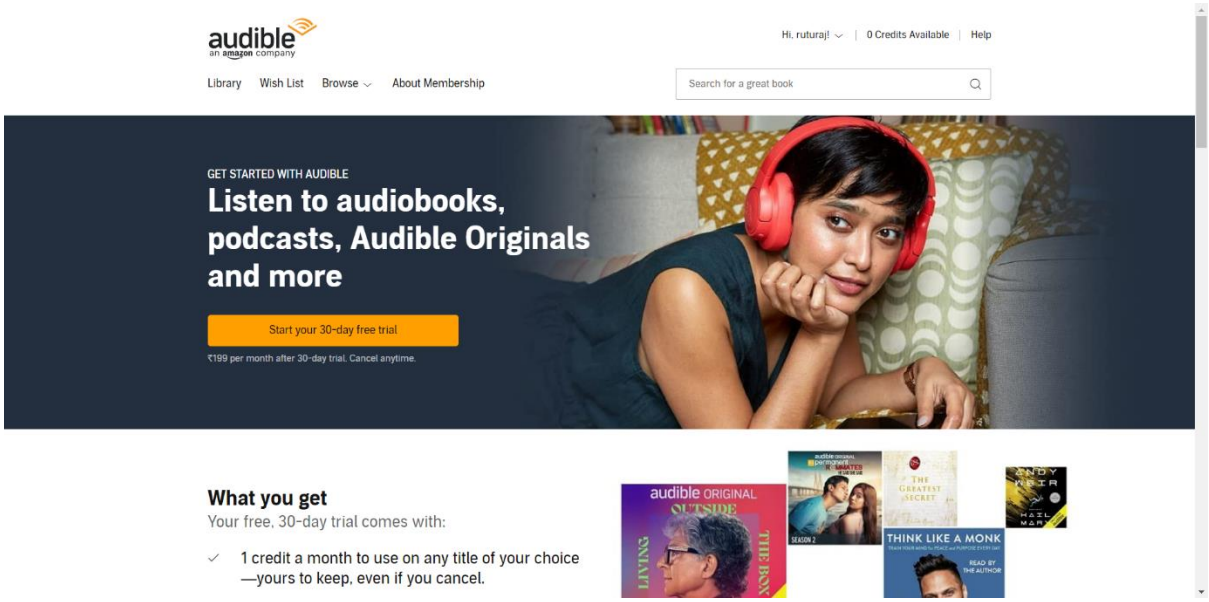
I have been working as a Quality Assurance Engineer Intern (Automation Tester) on this project throughout this Internship Program.

RESPONSIBLITIES OF A QUALITY ASSURANCE ENGINEER:

- Developing and implementing test plans and test cases to ensure software products meet quality standards and functional requirements.
- Identifying, documenting, and tracking defects and issues found during testing, and working with development teams to resolve them.
- Collaborating with software developers to ensure that software products are designed and built to be testable and maintainable.
- Developing and maintaining automated testing scripts and tools to increase testing efficiency and coverage.
- Conducting code reviews and providing feedback to developers to improve code quality and maintainability.
- Participating in software design and requirements review meetings to provide input and ensure that testing requirements are considered.
- Participating in the development and maintenance of test environments and data sets to support testing activities.
- Analyzing and interpreting test results and providing reports to stakeholders on the quality and readiness of software products for release.
- Continuously improving testing processes and methodologies to increase efficiency and effectiveness of testing activities.
- Staying up to date with industry trends and best practices in software testing and quality assurance.

PROJECT UNDERTAKEN DURING INTERNSHIP TRAINING:

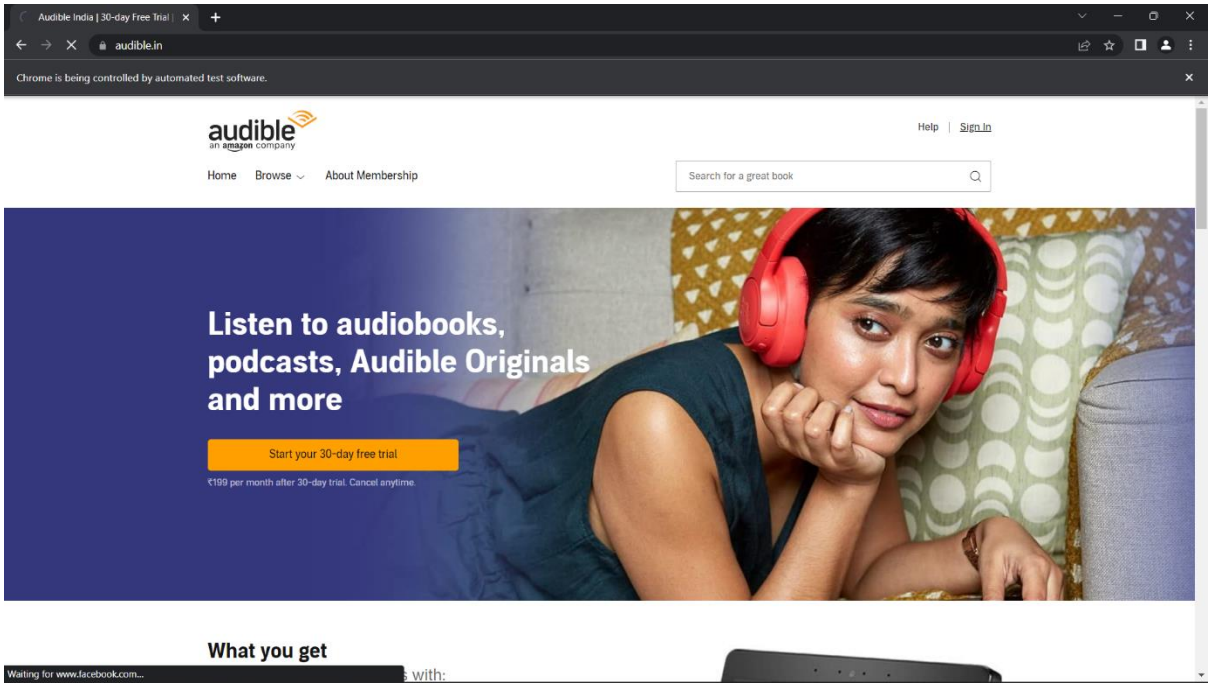
TESTING THE AMAZON AUDIBLE WEBISITE



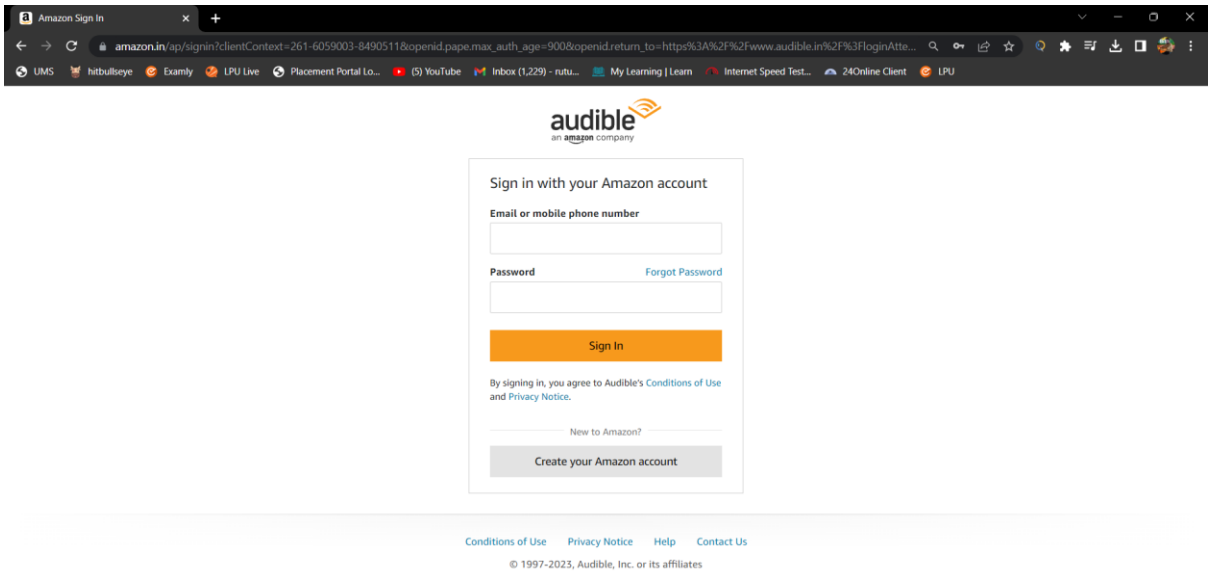
OBJECTIVES OF THE WORK UNDERTAKEN:

- To test the login functionality of the Amazon Audible website using Selenium WebDriver and Java.
- To verify that the login page displays an error message when an empty login is submitted.
- To verify that the login page displays an error message when invalid credentials are submitted.
- To verify that the user is able to login successfully using valid credentials.
- To test the browsing functionality of the Amazon Audible website using Selenium WebDriver and Java.
- To verify that the user is able to select a book and add it to their cart.
- To test the checkout functionality of the Amazon Audible website using Selenium WebDriver and Java.
- To verify that the user is able to complete the purchase of the selected book.
- To report any issues or bugs encountered during the testing process.
- To provide recommendations for improving the website's functionality and user experience based on the testing results.

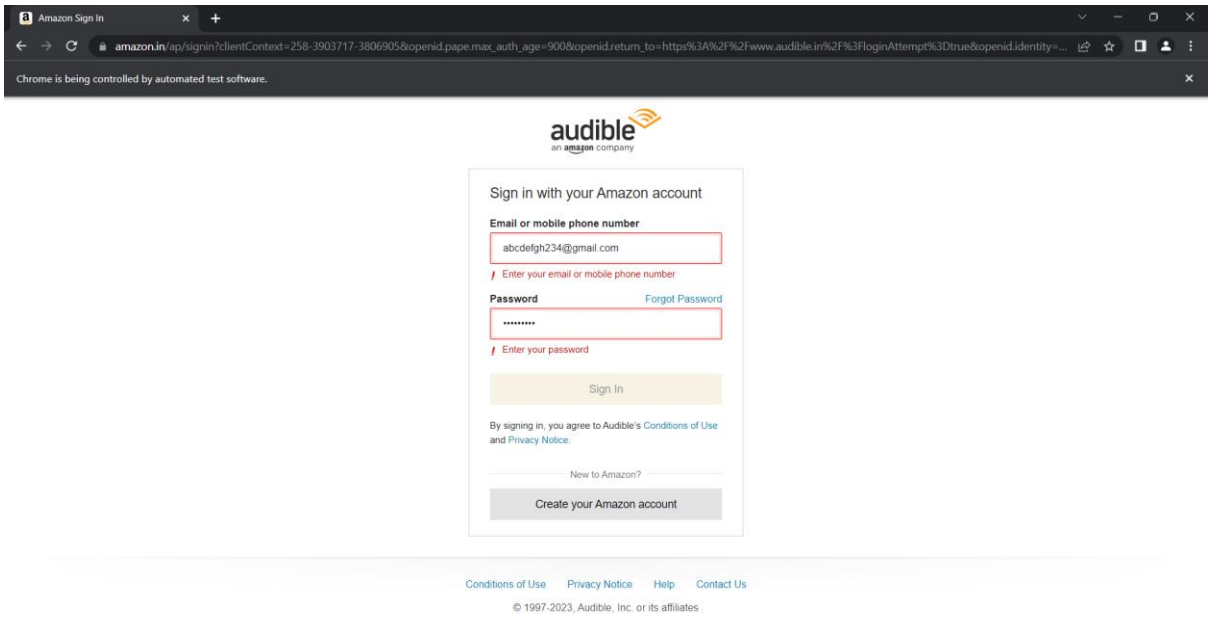
PROJECT WORK AND THE CODE SNIPPETS



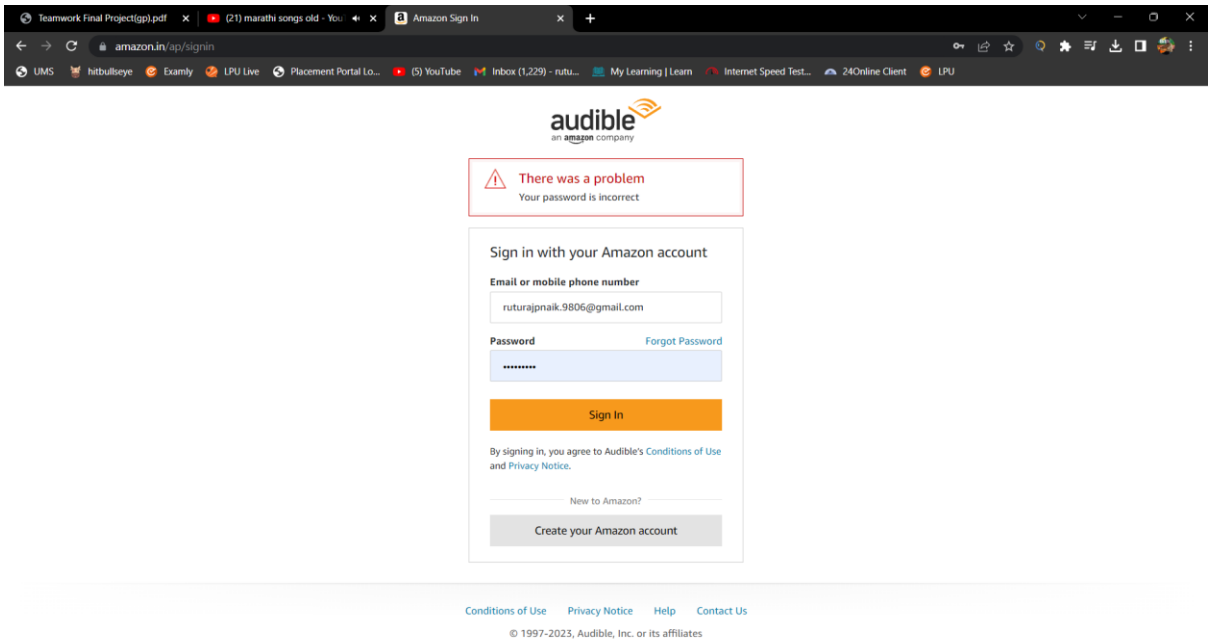
SIGN IN WITHOUT CREDENTIALS

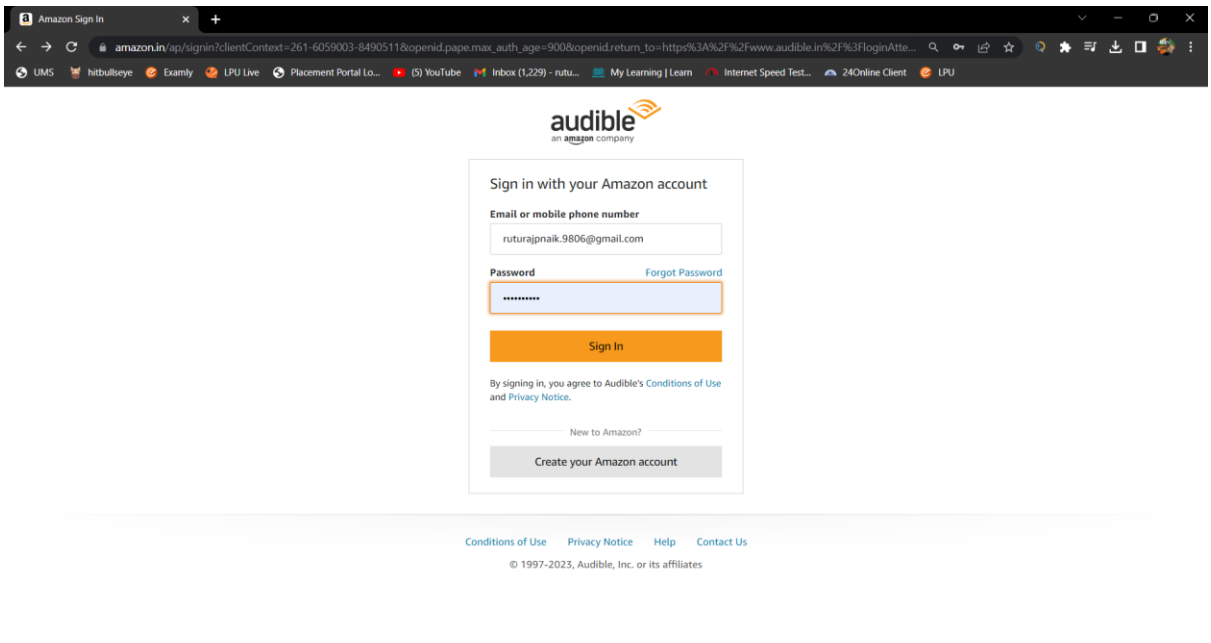


SIGN IN WITH WRONG CREDENTIALS

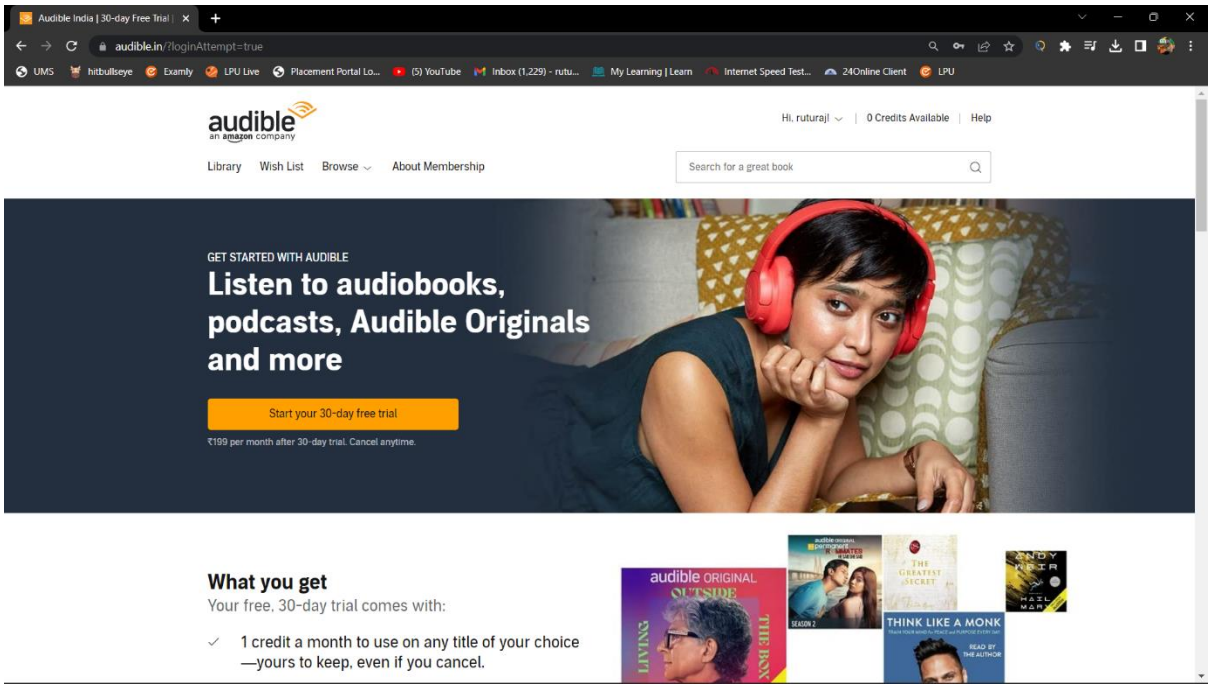


SIGN IN WITH RIGHT CREDENTIALS

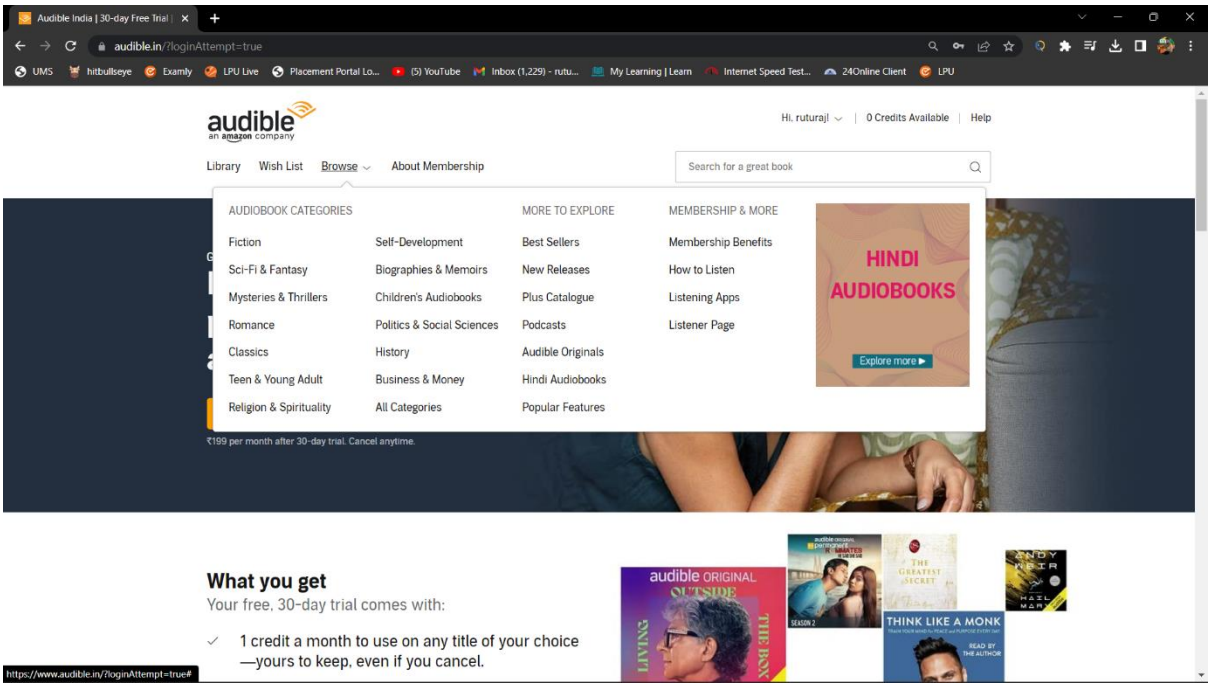




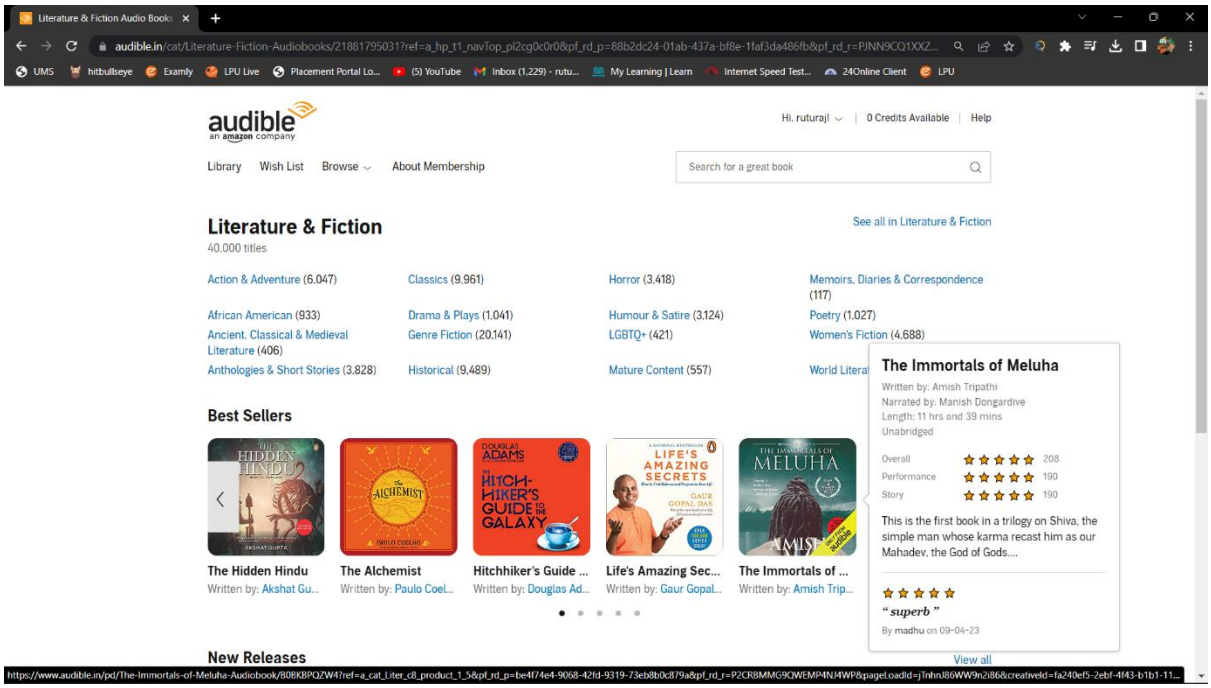
SELECTING THE BROWSE OPTION TO OPEN BOOK CATEGORIES



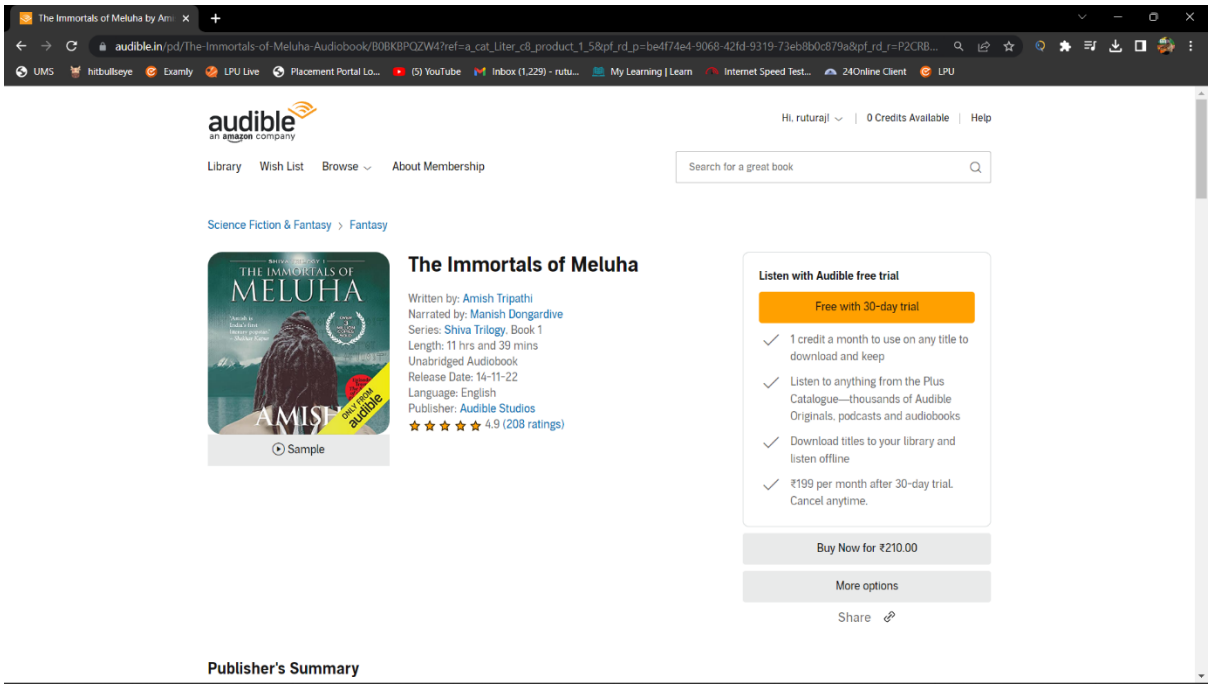
SELECTING SCI – FI & FANTASY CATEGORY OF BOOK



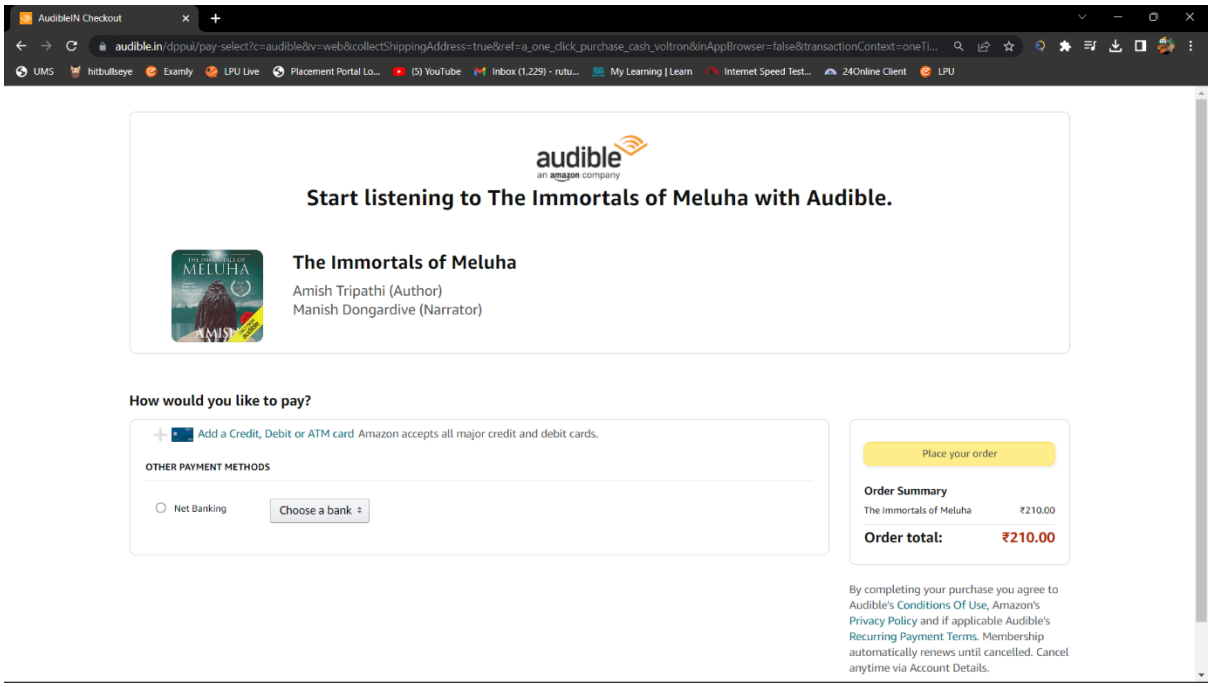
SELECTING THE BOOK NAMED THE IMMORTALS OF MELUHA



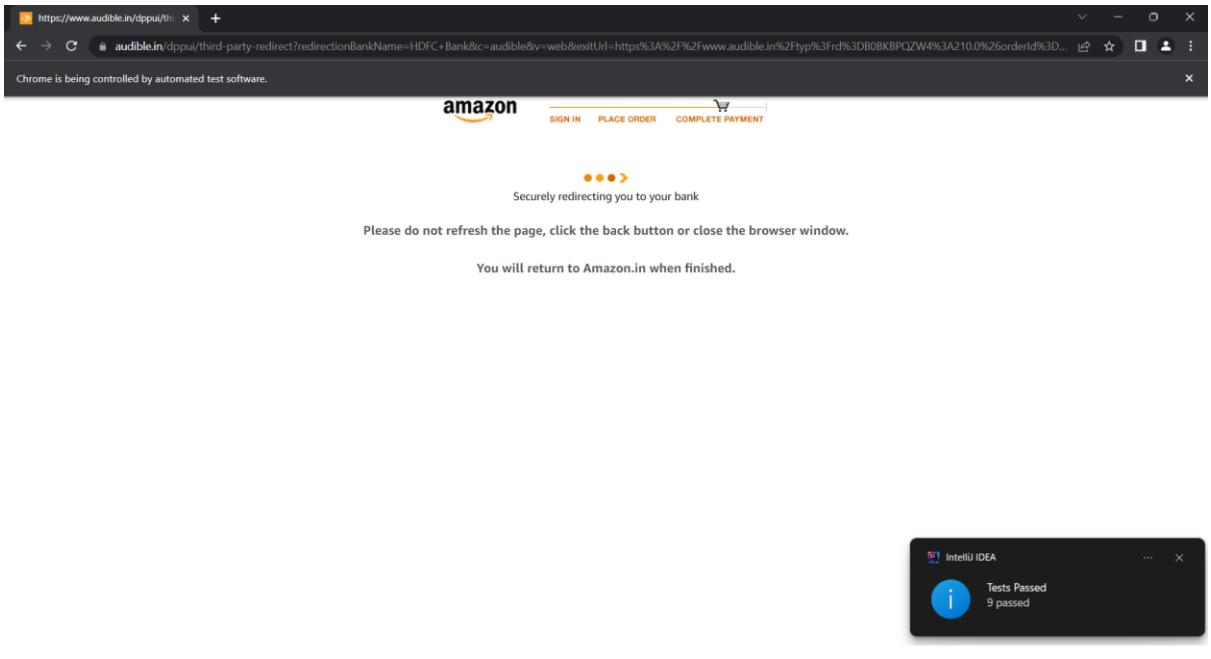
SELECTING 'BUY NOW FOR 210.00rs ' TO BUY THE BOOK



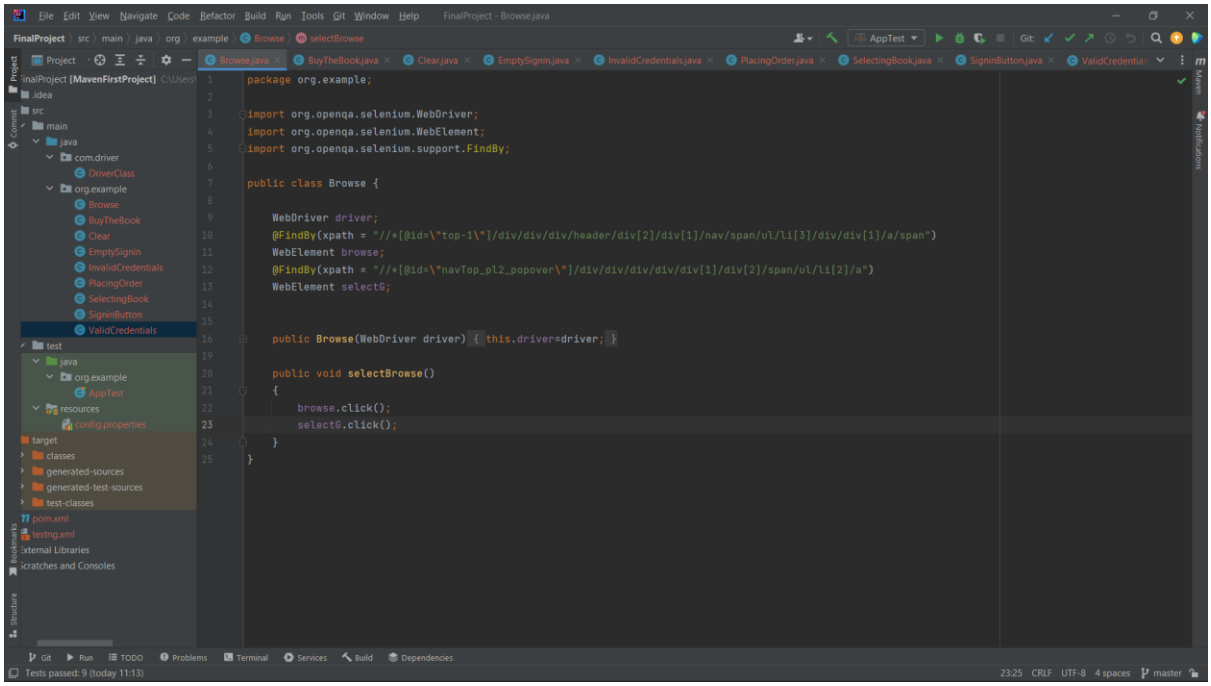
SELECTING NET BANKING AND BANK AND THEN PLACE YOUR ORDER TO BUY THE BOOK



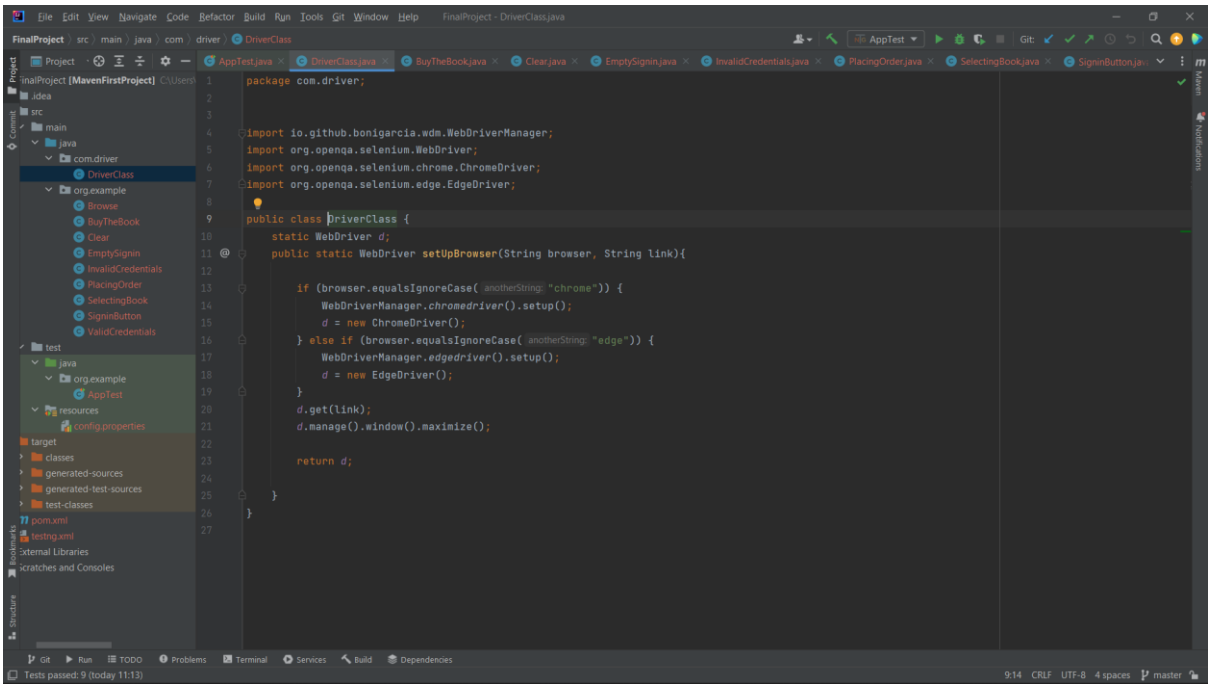
SUCCESSFULLY ALL THE TEST CASES GET PASSED



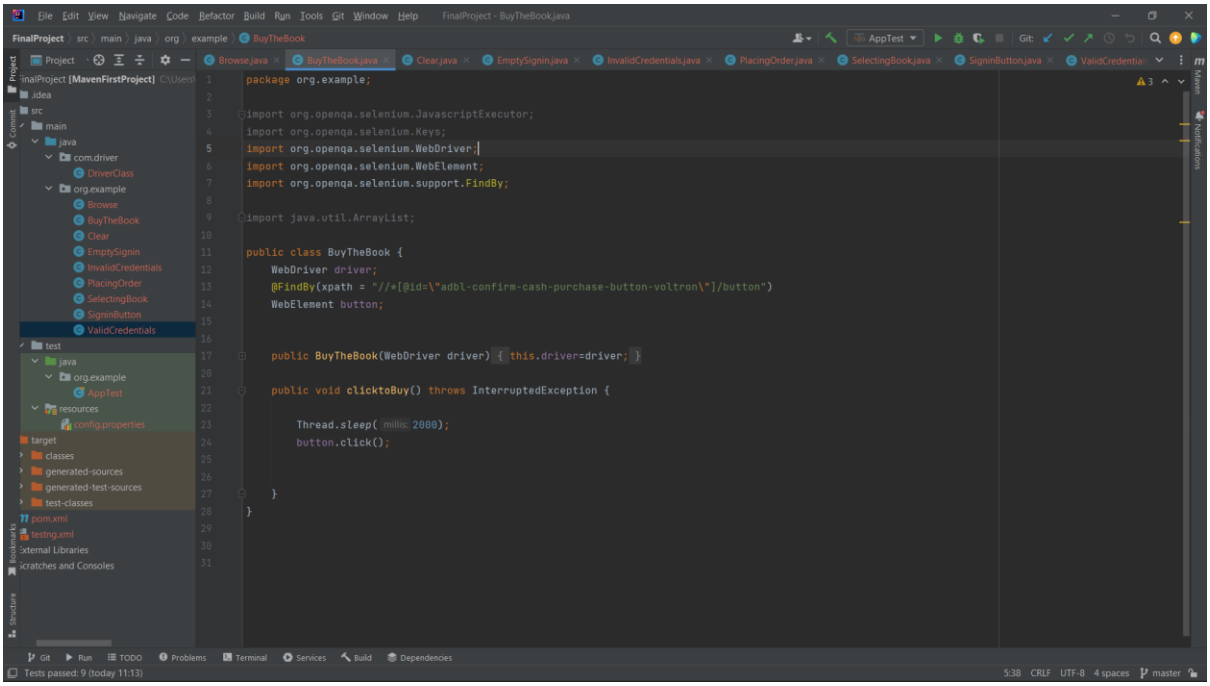
SNIPPETS OF THE CODE



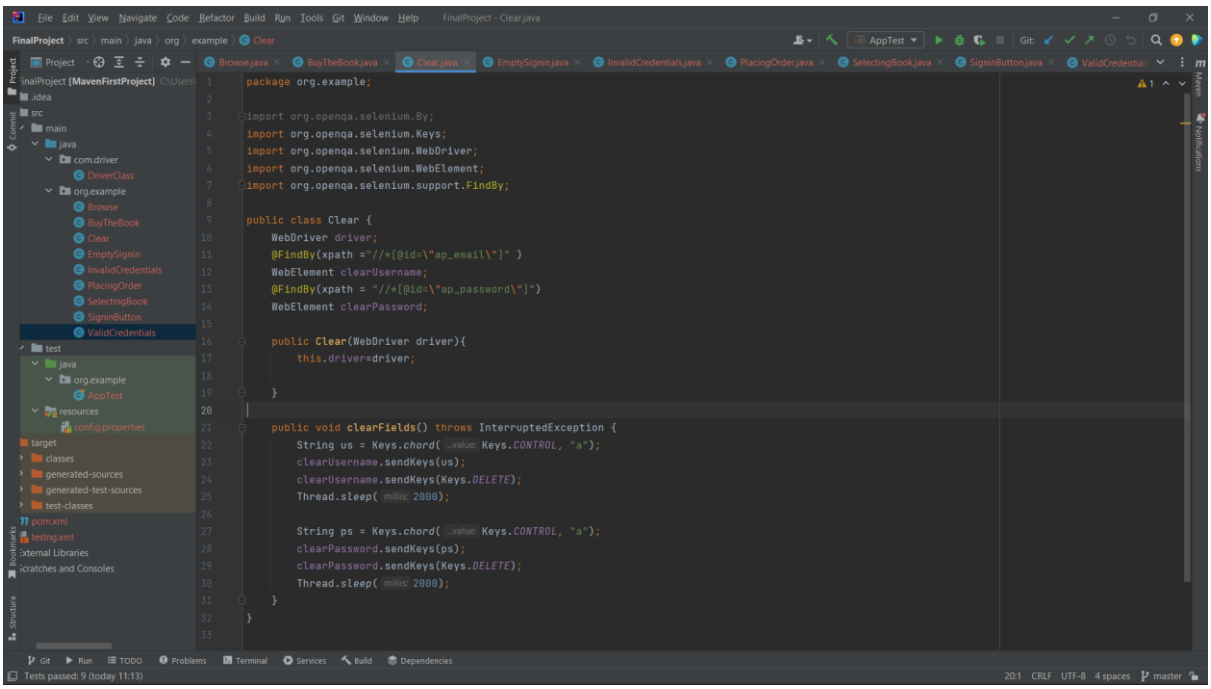
```java
package org.example;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;

public class Browse {

    WebDriver driver;
    @FindBy(xpath = "//*[@id=\"top-1\"]/div/div/div/header/div[2]/div[1]/nav/span/ul/li[3]/div/div[1]/a/span")
    WebElement browse;
    @FindBy(xpath = "//*[@id=\"navTop_pl2_popover\"]/div/div/div/div/div[1]/div[2]/span/ul/li[2]/a")
    WebElement selectG;


    public Browse(WebDriver driver) { this.driver=driver; }

    public void selectBrowse()
    {
        browse.click();
        selectG.click();
    }
}
```



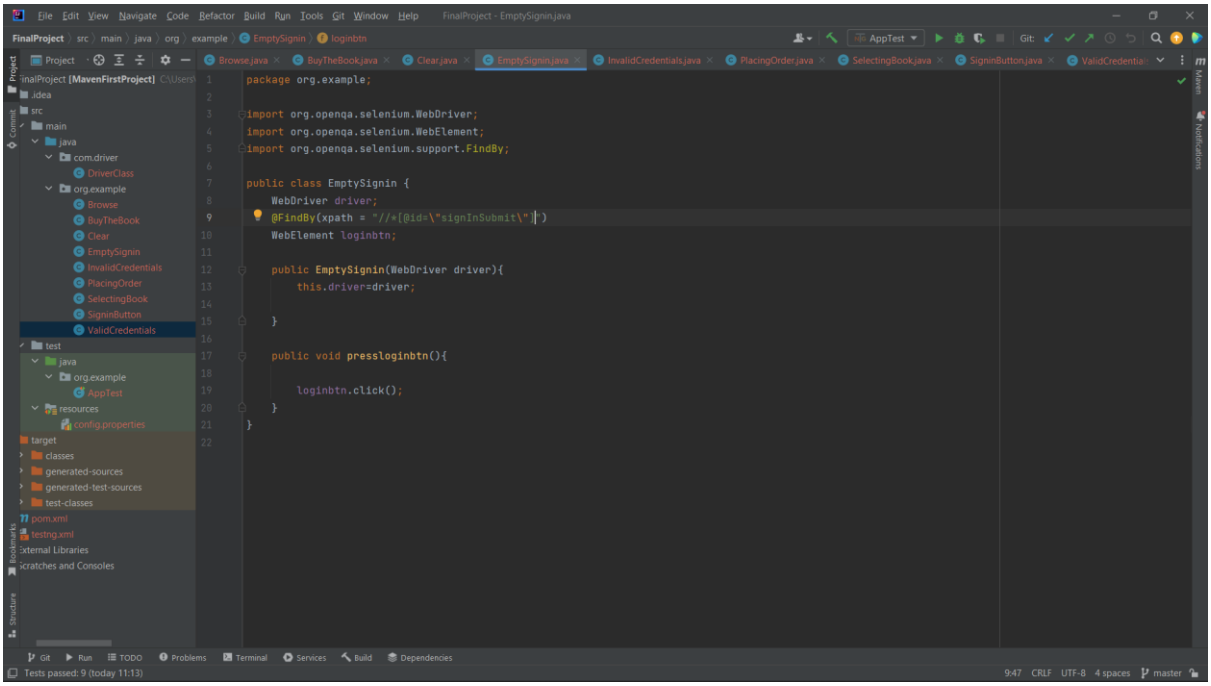
```java
package com.driver;


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 DriverClass {
    static WebDriver d;
    public static WebDriver setUpBrowser(String browser, String link){

        if (browser.equalsIgnoreCase( anotherString: "chrome")) {
            WebDriverManager.chromedriver().setup();
            d = new ChromeDriver();
        } else if (browser.equalsIgnoreCase( anotherString: "edge")) {
            WebDriverManager.edgedriver().setup();
            d = new EdgeDriver();
        }
        d.get(link);
        d.manage().window().maximize();

        return d;

    }
}
```



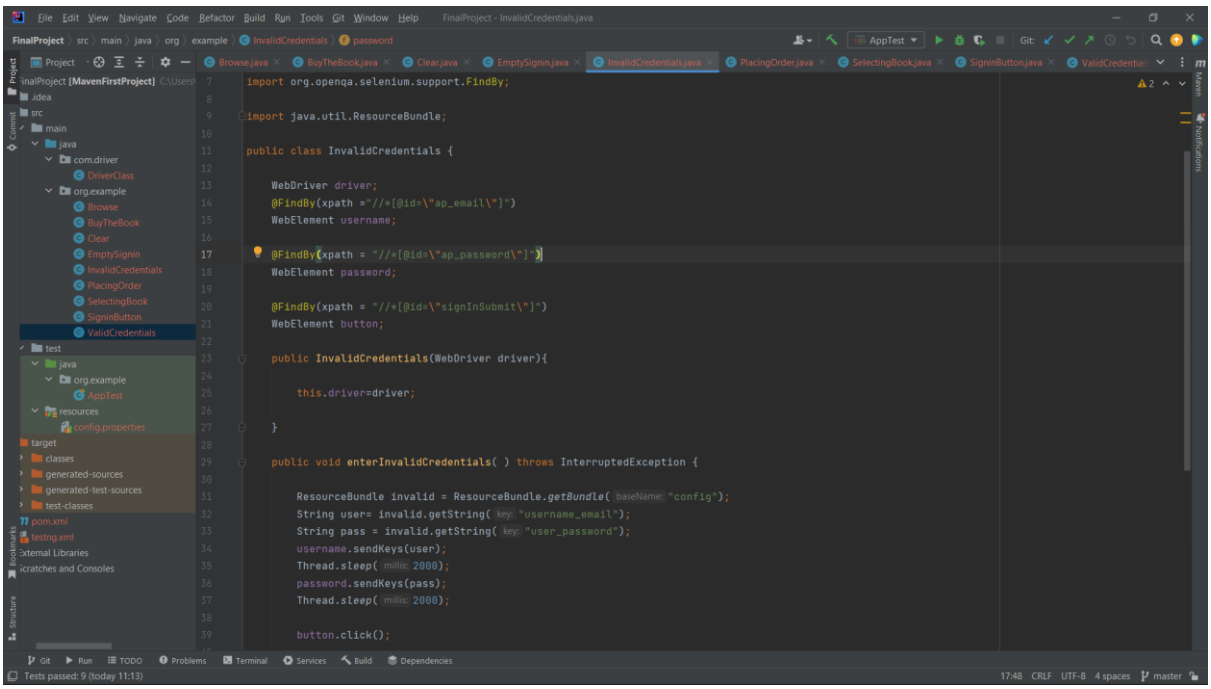
```java
package org.example;

import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;

import java.util.ArrayList;

public class BuyTheBook {
    WebDriver driver;
    @FindBy(xpath = "//*[@id=\"adbl-confirm-cash-purchase-button-voltron\"]/button")
    WebElement button;


    public BuyTheBook(WebDriver driver) { this.driver=driver; }

    public void clicktoBuy() throws InterruptedException {

        Thread.sleep( millis: 2000);
        button.click();


    }
}
```



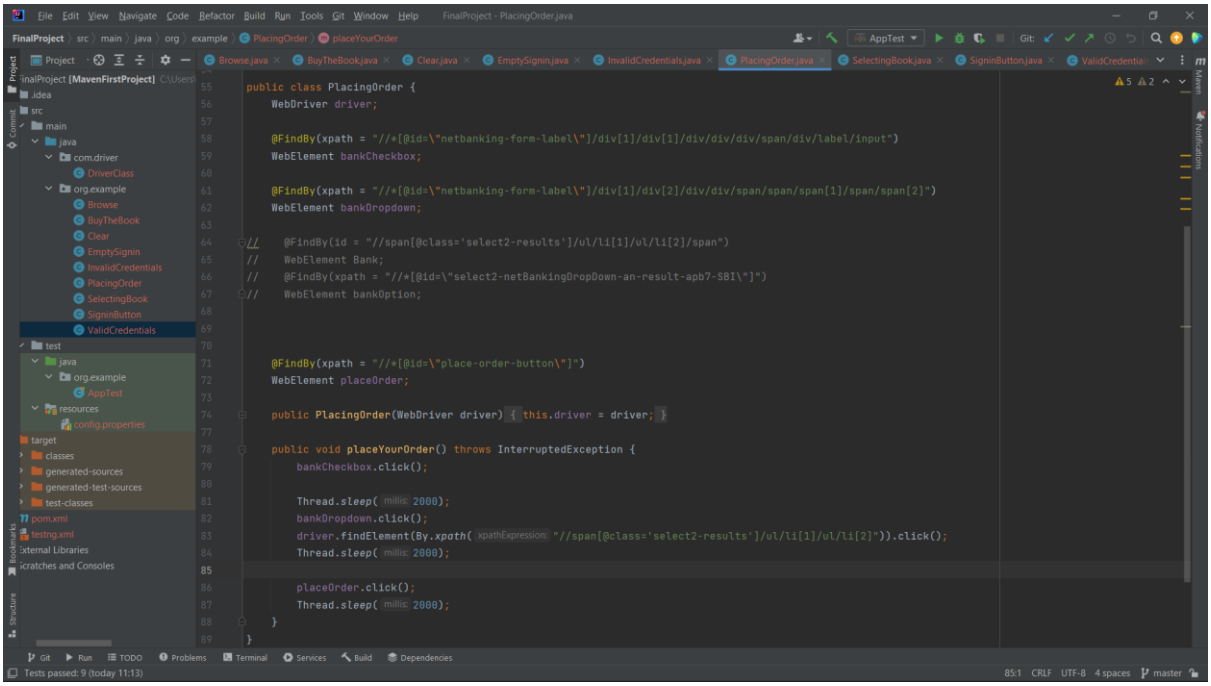
```java
package org.example;

import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;

public class Clear {
    WebDriver driver;
    @FindBy(xpath ="//*[@id=\"ap_email\"]" )
    WebElement clearUsername;
    @FindBy(xpath = "//*[@id=\"ap_password\"]")
    WebElement clearPassword;

    public Clear(WebDriver driver){
        this.driver=driver;


    }

    public void clearFields() throws InterruptedException {
        String us = Keys.chord( ...value: Keys.CONTROL, "a");
        clearUsername.sendKeys(us);
        clearUsername.sendKeys(Keys.DELETE);
        Thread.sleep( millis: 2000);

        String ps = Keys.chord( ...value: Keys.CONTROL, "a");
        clearPassword.sendKeys(ps);
        clearPassword.sendKeys(Keys.DELETE);
        Thread.sleep( millis: 2000);
    }
}
```



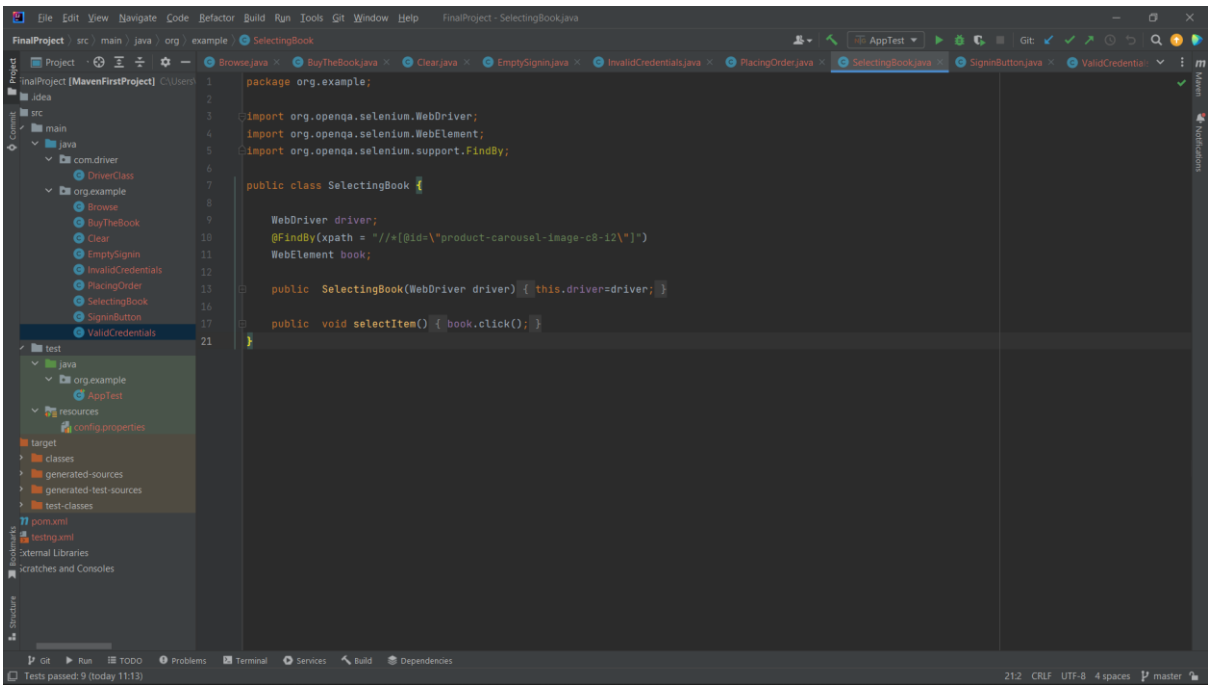
```java
package org.example;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;

public class EmptySignin {
    WebDriver driver;
    @FindBy(xpath = "//*[@id=\"signInSubmit\"]")
    WebElement loginbtn;

    public EmptySignin(WebDriver driver){
        this.driver=driver;

    }

    public void pressloginbtn(){

        loginbtn.click();
    }
}
```



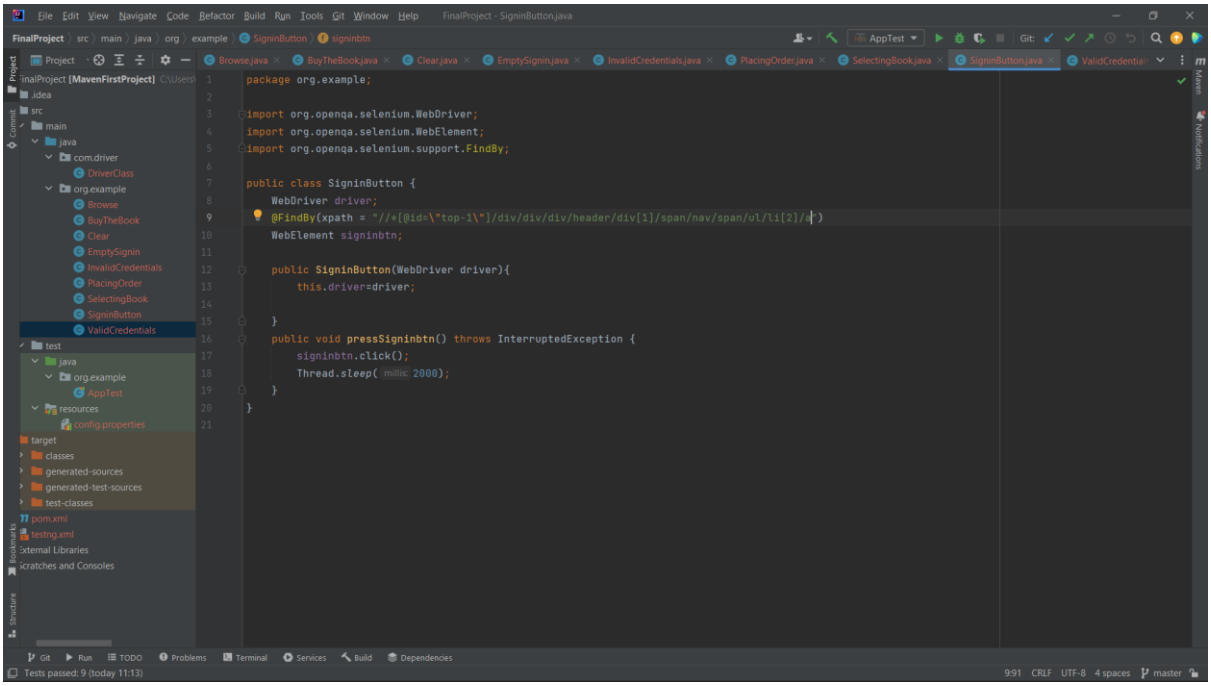
```java
import org.openqa.selenium.support.FindBy;

import java.util.ResourceBundle;

public class InvalidCredentials {

    WebDriver driver;
    @FindBy(xpath ="//*[@id=\"ap_email\"]")
    WebElement username;

    @FindBy(xpath = "//*[@id=\"ap_password\"]")
    WebElement password;

    @FindBy(xpath = "//*[@id=\"signInSubmit\"]")
    WebElement button;

    public InvalidCredentials(WebDriver driver){

        this.driver=driver;


    }

    public void enterInvalidCredentials( ) throws InterruptedException {

        ResourceBundle invalid = ResourceBundle.getBundle( baseName: "config");
        String user= invalid.getString( key: "username_email");
        String pass = invalid.getString( key: "user_password");
        username.sendKeys(user);
        Thread.sleep( millis: 2000);
        password.sendKeys(pass);
        Thread.sleep( millis: 2000);

        button.click();
```



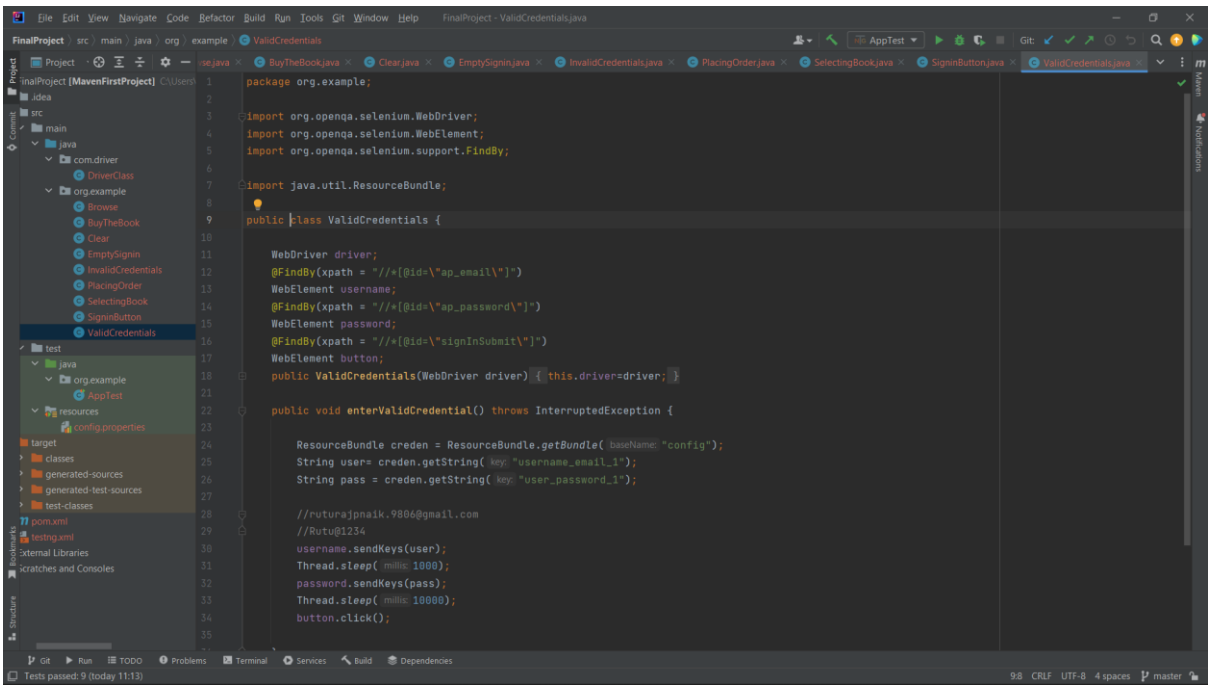
```java
public class PlacingOrder {
    WebDriver driver;

    @FindBy(xpath = "//*[@id=\"netbanking-form-label\"]/div[1]/div[1]/div/div/div/span/div/label/input")
    WebElement bankCheckbox;

    @FindBy(xpath = "//*[@id=\"netbanking-form-label\"]/div[1]/div[2]/div/div/span/span/span[1]/span/span[2]")
    WebElement bankDropdown;

//    @FindBy(id = "//span[@class='select2-results']/ul/li[1]/ul/li[2]/span")
//    WebElement Bank;
//    @FindBy(xpath = "//*[@id=\"select2-netBankingDropDown-an-result-apb7-SBI\"]")
//    WebElement bankOption;



    @FindBy(xpath = "//*[@id=\"place-order-button\"]")
    WebElement placeOrder;

    public PlacingOrder(WebDriver driver) { this.driver = driver; }

    public void placeYourOrder() throws InterruptedException {
        bankCheckbox.click();

        Thread.sleep( millis: 2000);
        bankDropdown.click();
        driver.findElement(By.xpath( xpathExpression: "//span[@class='select2-results']/ul/li[1]/ul/li[2]")).click();
        Thread.sleep( millis: 2000);

        placeOrder.click();
        Thread.sleep( millis: 2000);
    }
}
```



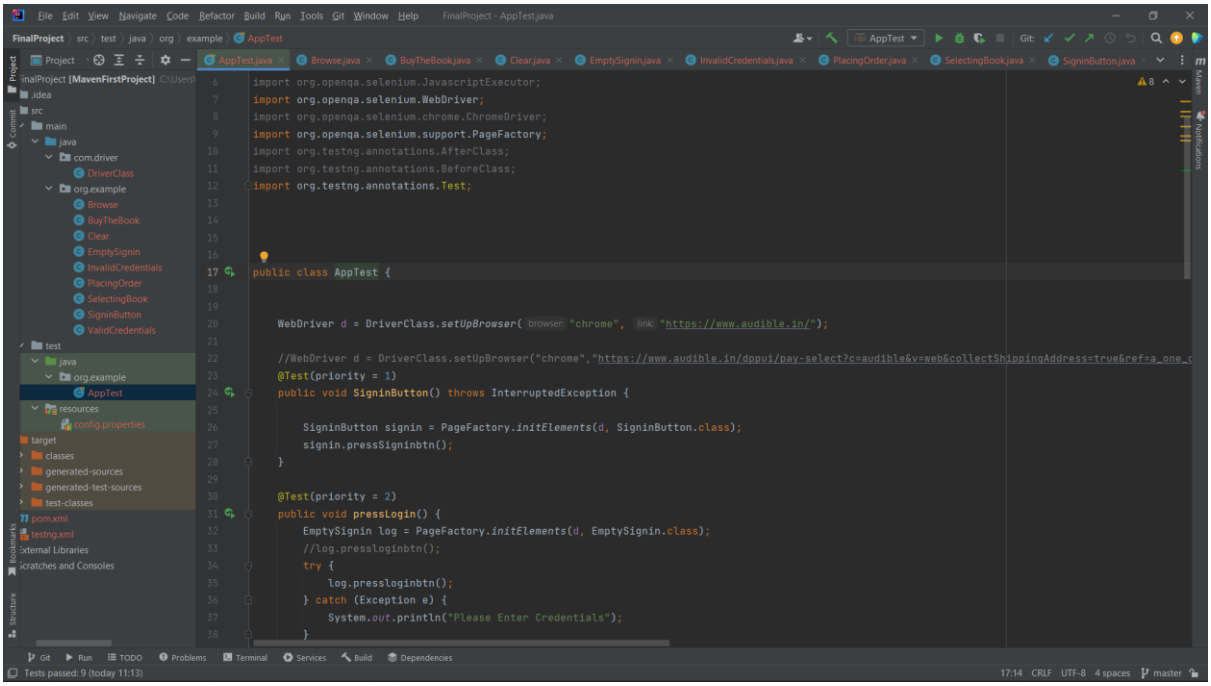
```java
package org.example;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;

public class SelectingBook {

    WebDriver driver;
    @FindBy(xpath = "//*[@id=\"product-carousel-image-c8-i2\"]")
    WebElement book;

    public  SelectingBook(WebDriver driver) { this.driver=driver; }

    public  void selectItem() { book.click(); }
}
```



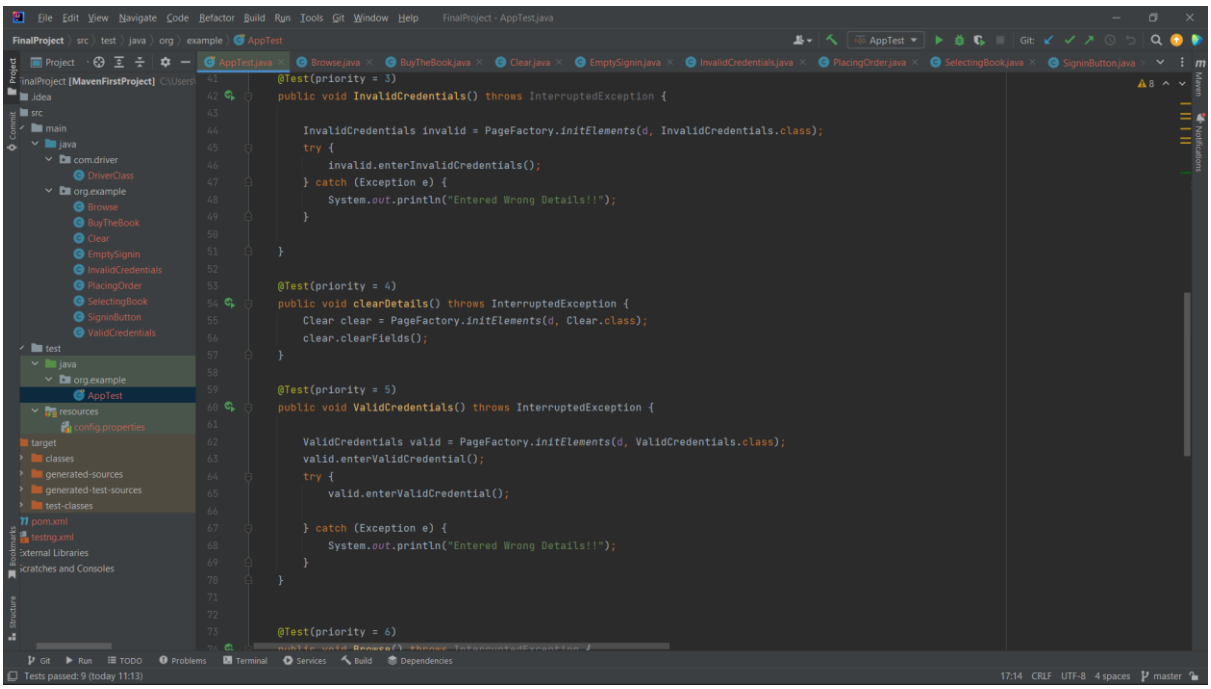
```java
package org.example;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;

public class SigninButton {
    WebDriver driver;
    @FindBy(xpath = "//*[@id=\"top-1\"]/div/div/div/header/div[1]/span/nav/span/ul/li[2]/a")
    WebElement signinbtn;

    public SigninButton(WebDriver driver){
        this.driver=driver;

    }
    public void pressSigninbtn() throws InterruptedException {
        signinbtn.click();
        Thread.sleep( millis: 2000);
    }
}
```



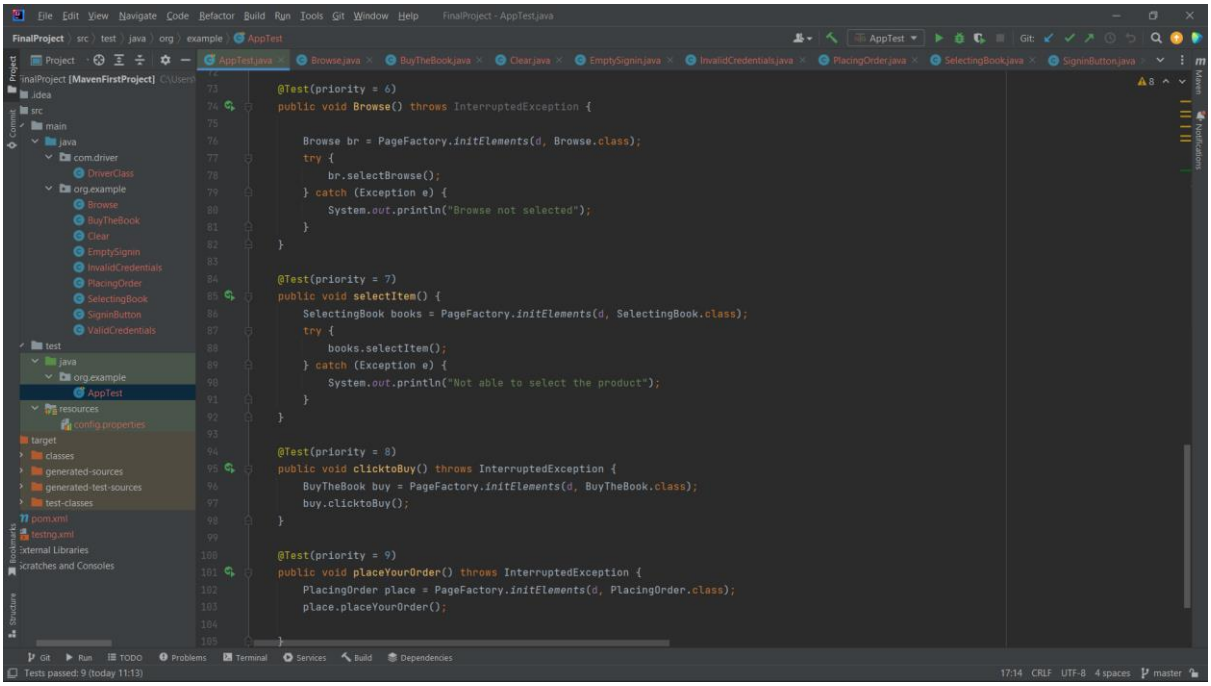
```java
package org.example;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;

import java.util.ResourceBundle;

public class ValidCredentials {

    WebDriver driver;
    @FindBy(xpath = "//*[@id=\"ap_email\"]")
    WebElement username;
    @FindBy(xpath = "//*[@id=\"ap_password\"]")
    WebElement password;
    @FindBy(xpath = "//*[@id=\"signInSubmit\"]")
    WebElement button;
    public ValidCredentials(WebDriver driver) { this.driver=driver; }

    public void enterValidCredential() throws InterruptedException {

        ResourceBundle creden = ResourceBundle.getBundle( baseName: "config");
        String user= creden.getString( key: "username_email_1");
        String pass = creden.getString( key: "user_password_1");

        //ruturajpnaik.9806@gmail.com
        //Rutu@1234
        username.sendKeys(user);
        Thread.sleep( millis: 1000);
        password.sendKeys(pass);
        Thread.sleep( millis: 10000);
        button.click();
```



```
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.PageFactory;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;



public class AppTest {


    WebDriver d = DriverClass.setUpBrowser( browser: "chrome",  link: "https://www.audible.in/");

    //WebDriver d = DriverClass.setUpBrowser("chrome","https://www.audible.in/dppui/pay-select?c=audible&v=web&collectShippingAddress=true&ref=a_one_c
    @Test(priority = 1)
    public void SigninButton() throws InterruptedException {

        SigninButton signin = PageFactory.initElements(d, SigninButton.class);
        signin.pressSigninbtn();
    }

    @Test(priority = 2)
    public void pressLogin() {
        EmptySignin log = PageFactory.initElements(d, EmptySignin.class);
        //log.pressloginbtn();
        try {
            log.pressloginbtn();
        } catch (Exception e) {
            System.out.println("Please Enter Credentials");
        }
```



```
    @Test(priority = 3)
    public void InvalidCredentials() throws InterruptedException {

        InvalidCredentials invalid = PageFactory.initElements(d, InvalidCredentials.class);
        try {
            invalid.enterInvalidCredentials();
        } catch (Exception e) {
            System.out.println("Entered Wrong Details!!");
        }

    }

    @Test(priority = 4)
    public void clearDetails() throws InterruptedException {
        Clear clear = PageFactory.initElements(d, Clear.class);
        clear.clearFields();
    }

    @Test(priority = 5)
    public void ValidCredentials() throws InterruptedException {

        ValidCredentials valid = PageFactory.initElements(d, ValidCredentials.class);
        valid.enterValidCredential();
        try {
            valid.enterValidCredential();

        } catch (Exception e) {
            System.out.println("Entered Wrong Details!!");
        }
    }


    @Test(priority = 6)
```


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
FinalProject - AppTest.java
FinalProject [MavenFirstProject]
.idea
src
main
java
com.driver
DriverClass
org.example
Browse
BuyTheBook
Clear
EmptySignin
InvalidCredentials
PlacingOrder
SelectingBook
SigninButton
ValidCredentials
test
java
org.example
AppTest
resources
config.properties
target
classes
generated-sources
generated-test-sources
test-classes
pom.xml
testng.xml
External Libraries
Scratches and Consoles
AppTest.java
Browse.java
BuyTheBook.java
Clear.java
EmptySignin.java
InvalidCredentials.java
PlacingOrder.java
SelectingBook.java
SigninButton.java
@Test(priority = 6)
public void Browse() throws InterruptedException {
Browse br = PageFactory.initElements(d, Browse.class);
try {
br.selectBrowse();
} catch (Exception e) {
System.out.println("Browse not selected");
}
}
@Test(priority = 7)
public void selectItem() {
SelectingBook books = PageFactory.initElements(d, SelectingBook.class);
try {
books.selectItem();
} catch (Exception e) {
System.out.println("Not able to select the product");
}
}
@Test(priority = 8)
public void clicktoBuy() throws InterruptedException {
BuyTheBook buy = PageFactory.initElements(d, BuyTheBook.class);
buy.clicktoBuy();
}
@Test(priority = 9)
public void placeYourOrder() throws InterruptedException {
PlacingOrder place = PageFactory.initElements(d, PlacingOrder.class);
place.placeYourOrder();
}
Git Run TODO Problems Terminal Services Build Dependencies
Tests passed: 9 (today 11:13)
17:14 CRLF UTF-8 4 spaces master