INTERNSHIP REPORT

Name: Nivedha Vasudevan

Internship Organisation: NULLCLASS EDTECH PRIVATE LIMITED

Duration: 26 April 2025 - 26 May 2025

Role: Software Tester Intern

INTRODUCTION:

The report outlines my one month internship at NullClass, where I worked on selenium based web automation tasks. This work focuses on practical implementation of automation scripts, testing and scraping techniques within specified time constraints.

BACKGROUND:

Null class provides training session about Introduction to Selenium Testing which included testing fundamentals, Test automation, Selenium Components, core java and Testing on Real Amazon Website which includes WebDriver with project testing, Framework design and implementation, Implementing Testing on amazon website. During my internship period I have developed end to end selenium scripts under certain conditions like time based execution and element selection.

LEARNING OBJECTIVES:

- Learned to apply Webdriver for browser automation.
- Developed reusable test scripts for function testing.
- Implemented practices like time based conditions, exception handling etc,.
- Enhancing bugs and validation techniques by java and chrome tools.

ACTIVITIES AND TASKS:

Task 1: Selenium code to open a browser and navigate to a URL.

- Used WebDriver to setup chrome driver
- Opened chrome and navigated to the site https://www.google.com/
- Added sleep and printed a success message.

Task 2: Automate the process of searching for a product.

- Automated product search with various filters like price > ₹2000, rating > 4, brand starts with 'C'
- The code runs only between 3 pm to 6 pm
- Used multiple css selectors for resilience.
- Implemented each filter in a modular way.

Task 3: Login and profile validation

- The code runs only between 12 pm to 3 pm.
- Used amazon website by manual user login.
- Extracted username from possible DOM locations and validated that it **does not contain** characters (A, C, G, I, L, K).
- Captured screenshots for both success and error cases.
- Closed browser and scanner after execution.

SKILLS AND COMPETENCIES:

Technical skills:

- Selenium WebDriver
- Java
- Xpath
- CSS Selectors
- Javascript Executor
- WebDriver Manager
- Exception Handling
- Time based task execution

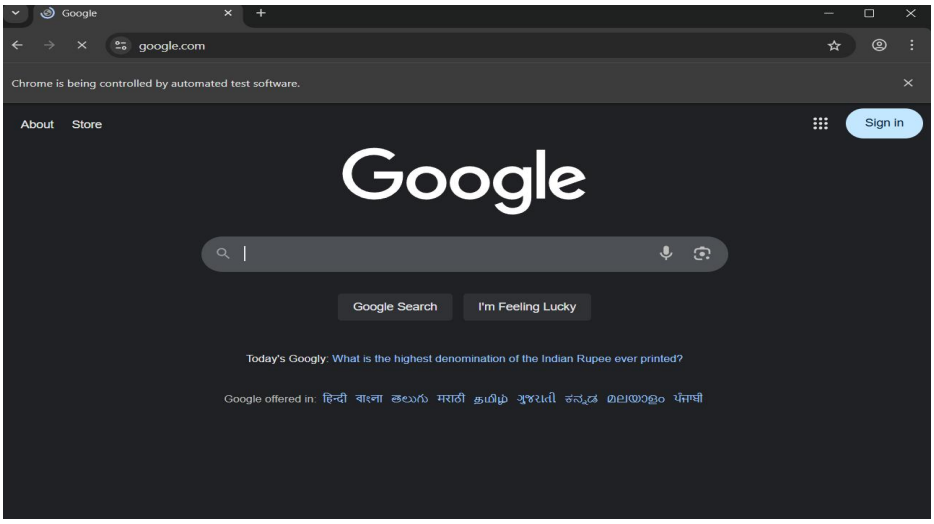
Soft skills:

- Problem Solving
- Time management
- Code Modularity

FEEDBACK AND EVIDENCE:

Task 1:

```java
package com.gvp.com.HibernateCRUDApp2;

import org.openqa.selenium.WebDriver;

public class ChromeDriverTest {

    public static void main(String[] args) {
        try {

            WebDriverManager.chromedriver().setup();


            WebDriver driver = new ChromeDriver();


            driver.get("https://www.google.com");

            System.out.println("Opened Google successfully!");


            Thread.sleep(2000);
            driver.quit();

        } catch (Exception e) {

            System.out.println("An error occurred: " + e.getMessage());
            e.printStackTrace();
        }
    }
}
```



Task 2:

```
        options.addArguments("--disable-notifications");
        options.addArguments("--disable-popup-blocking");
        options.addArguments("--disable-gpu");
        options.addArguments("--no-sandbox");
        options.addArguments("--disable-dev-shm-usage");
        options.addArguments("--ignore-certificate-errors");


        options.addArguments("user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like

        System.out.println("Initializing Chrome WebDriver...");
        return new ChromeDriver(options);
    }

    private static boolean isWithinTimeWindow(LocalTime currentTime) {
        return !currentTime.isBefore(START_TIME) && !currentTime.isAfter(END_TIME);
    }

    private static void navigateToAmazon(WebDriver driver) {
        System.out.println("Navigating to Amazon...");
        driver.get(TARGET_URL);
        driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(30));
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        if (driver.getTitle().contains("Amazon")) {
            System.out.println("Successfully loaded Amazon website");
        } else {
            System.out.println("Warning: Page title doesn't contain 'Amazon'. Title: " + driver.getTitle());
```

```
private static void searchForProduct(WebDriver driver, WebDriverWait wait, String query) {
    try {
        System.out.println("Looking for search box...");
        WebElement searchBox = null;

        try {
            searchBox = wait.until(ExpectedConditions.elementToBeClickable(By.id("twotabsearchtextbox")));
        } catch (TimeoutException e) {
            System.out.println("Could not find search box by ID. Trying alternate selectors...");
            try {
                searchBox = wait.until(ExpectedConditions.elementToBeClickable(
                        By.xpath("//input[@type='text' and @placeholder='Search Amazon.in']")));
            } catch (TimeoutException e2) {
                // Try JavaScript as last resort
                System.out.println("Using JavaScript to find search box...");
                JavascriptExecutor js = (JavascriptExecutor) driver;
                searchBox = (WebElement) js.executeScript(
                        "return document.querySelector('input[type=\"text\"][placeholder*=\"Search\"]');");
            }
        }

        if (searchBox == null) {
            throw new NoSuchElementException("Could not locate search box with any method");
        }

        searchBox.clear();
        searchBox.sendKeys(query);
        System.out.println("Submitting search query: " + query);
        searchBox.submit();
```

```
            try {
                wait.until(ExpectedConditions.or(
                    ExpectedConditions.presenceOfElementLocated(By.cssSelector("div.s-main-slot")),
                    ExpectedConditions.presenceOfElementLocated(By.cssSelector(".s-result-list")),
                    ExpectedConditions.titleContains(query)
                ));
                System.out.println("Search results loaded for: " + query);
            } catch (TimeoutException e) {
                System.out.println("Timed out waiting for search results, but continuing...");
            }

            sleep(2000);

        } catch (Exception e) {
            System.err.println("Error during search: " + e.getMessage());
        }
    }

    private static void applyPriceFilter(WebDriver driver, WebDriverWait wait, int minPrice) {
        System.out.println("Attempting to apply price filter (> ₹" + minPrice + ")");
        try {

            try {

                WebElement minPriceInput = wait.until(ExpectedConditions.elementToBeClickable(By.id("low-price")));
                minPriceInput.clear();
                minPriceInput.sendKeys(String.valueOf(minPrice));

                WebElement priceGo = driver.findElement(By.xpath("//input[@class='a-button-input' and @type='submit
```

```
                WebElement priceGo = driver.findElement(By.xpath("//input[@class='a-button-input' and @type='submit
                priceGo.click();
            } catch (Exception e1) {
                System.out.println("Standard price filter failed, trying alternative method...");

                try {

                    WebElement minPriceInput = driver.findElement(By.xpath("//input[contains(@placeholder,'Min')]")
                    minPriceInput.clear();
                    minPriceInput.sendKeys(String.valueOf(minPrice));

                    WebElement priceGo = driver.findElement(By.xpath("//span[contains(text(),'Go')]/parent::*"));
                    priceGo.click();
                } catch (Exception e2) {

                    System.out.println("Using JavaScript to apply price filter...");
                    JavascriptExecutor js = (JavascriptExecutor) driver;
                    js.executeScript(
                        "let inputs = document.querySelectorAll('input[placeholder*=\"Min\"]');" +
                        "if(inputs.length > 0) {" +
                        "  inputs[0].value = '" + minPrice + "';" +
                        "  let button = document.querySelector('span:contains(\"Go\")').parentElement;" +
                        "  if(button) button.click();" +
                        "}"
                    );
                }
            }

            sleep(3000);
```

```
            sleep(3000);
            System.out.println("Price filter applied successfully");

        } catch (Exception e) {
            System.err.println("All attempts to apply price filter failed: " + e.getMessage());
        }


    ivate static void applyRatingFilter(WebDriver driver, WebDriverWait wait, int minRating) {
        System.out.println("Attempting to apply rating filter (" + minRating + " stars & up)");
        try {

            boolean filterApplied = false;


            try {
                List<WebElement> ratingOptions = driver.findElements(
                    By.xpath("//section[contains(@aria-label,'stars')]//i[contains(@class,'star')]"));

                if (!ratingOptions.isEmpty()) {
                    // Find option closest to our desired rating
                    for (WebElement option : ratingOptions) {
                        String ariaLabel = option.getAttribute("aria-label");
                        if (ariaLabel != null && ariaLabel.contains(minRating + " Stars")) {
                            System.out.println("Found rating element: " + ariaLabel);
                            scrollToElement(driver, option);
                            option.click();
                            filterApplied = true;
                            break;
                        }
```

```
    } catch (Exception e) {
        System.out.println("First rating filter approach failed: " + e.getMessage());
    }


    if (!filterApplied) {
        try {
            List<WebElement> textOptions = driver.findElements(
                By.xpath("//span[contains(text),'" + minRating + " Stars & Up')]"));

            if (!textOptions.isEmpty()) {
                WebElement ratingElement = textOptions.get(0);
                scrollToElement(driver, ratingElement);
                ratingElement.click();
                filterApplied = true;
            }
        } catch (Exception e) {
            System.out.println("Second rating filter approach failed: " + e.getMessage());
        }
    }

    if (!filterApplied) {
        try {
            System.out.println("Using JavaScript to apply rating filter...");
            JavascriptExecutor js = (JavascriptExecutor) driver;
            js.executeScript(
                "let ratings = document.querySelectorAll('*');" +
                "for(let i=0; i<ratings.length; i++) {" +
                "  let el = ratings[i];" +
                "  if(el.textContent && el.textContent.includes('" + minRating + " Stars & Up')) {" +
                "    el.click();" +
                "    return true;" +
                "  }" +
                "}" +
                "return false;"
            );
            filterApplied = true;
        } catch (Exception e) {
```

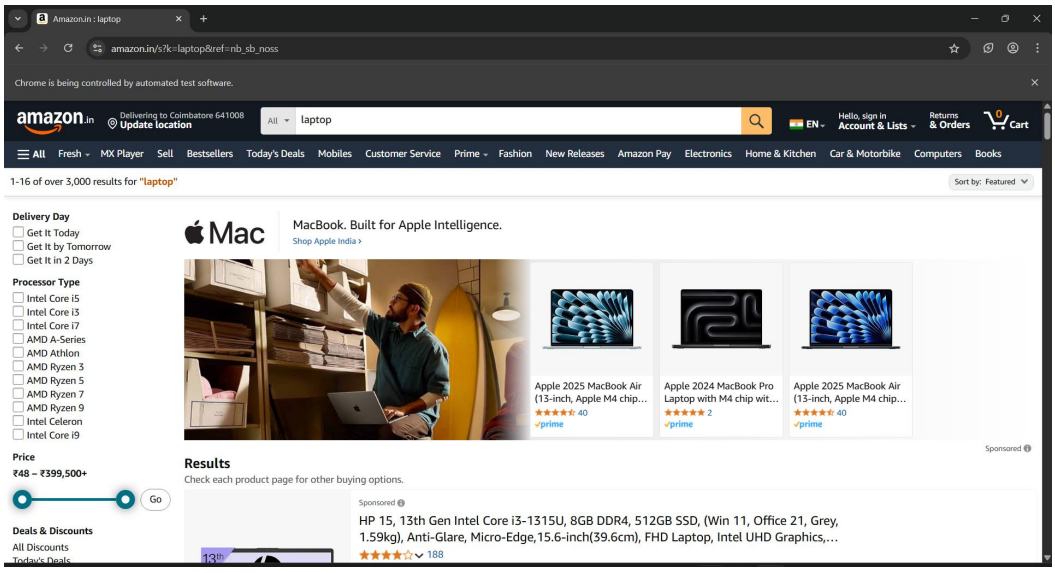
```
            System.out.println("JavaScript rating filter approach failed: " + e.getMessage());
        }
    }

    if (filterApplied) {

        sleep(3000);
        System.out.println("Rating filter applied successfully");
    } else {
        System.out.println("Could not apply rating filter with any method");
    }

} catch (Exception e) {
    System.err.println("Error applying rating filter: " + e.getMessage());
}


ivate static void displayFilteredResults(WebDriver driver, WebDriverWait wait, int maxResults) {
 System.out.println("\nGathering product information...");
 try {

    sleep(2000);
    scrollPage(driver);


    List<WebElement> productTitles = null;
    try {
        productTitles = driver.findElements(By.cssSelector("h2 a span"));
        if (productTitles.isEmpty()) {
            productTitles = driver.findElements(By.cssSelector(".a-size-medium.a-color-base"));
        }
        if (productTitles.isEmpty()) {
            productTitles = driver.findElements(By.cssSelector("[data-cy='title-recipe'] h2"));
        }
    } catch (Exception e) {
        System.out.println("Standard methods to find products failed, trying JavaScript...");
        JavascriptExecutor js = (JavascriptExecutor) driver;
        js.executeScript(
```

```
                productTitles.subList(0, maxResults) : productTitles;

        for (int i = 0; i < displayTitles.size(); i++) {
            String title = displayTitles.get(i).getText();

            String price = "Price not available";
            try {

                WebElement container = getParentContainer(driver, displayTitles.get(i));
                if (container != null) {
                    List<WebElement> priceElements = container.findElements(
                        By.cssSelector(".a-price-whole, .a-color-price"));
                    if (!priceElements.isEmpty()) {
                        price = priceElements.get(0).getText().replaceAll("[^0-9.,]", "");
                    }
                }
            } catch (Exception e) {

            }

            System.out.printf("%d. %-70s ₹%s%n", (i + 1),
                    title.length() > 70 ? title.substring(0, 67) + "..." : title,
                    price);
        }
    } else {

        for (int i = 0; i < Math.min(maxResults, filteredTitles.size()); i++) {
            String title = filteredTitles.get(i).getText();

            String price = "Price not available";
            try {

                WebElement container = getParentContainer(driver, filteredTitles.get(i));
                if (container != null) {
                    List<WebElement> priceElements = container.findElements(
                        By.cssSelector(".a-price-whole, .a-color-price"));
                    if (!priceElements.isEmpty()) {
                        price = priceElements.get(0).getText().replaceAll("[^0-9.,]", "");
```

```
            } catch (Exception e) {

            }

            System.out.printf("%d. %-70s ₹%s%n", (i + 1),
                    title.length() > 70 ? title.substring(0, 67) + "..." : title,
                    price);
        }
    }
} catch (Exception e) {
    System.err.println("Failed to display results: " + e.getMessage());
}


ivate static void sleep(long millis) {
try {
    Thread.sleep(millis);
} catch (InterruptedException e) {
    Thread.currentThread().interrupt();
}


ivate static void scrollPage(WebDriver driver) {
try {
    JavascriptExecutor js = (JavascriptExecutor) driver;

    for (int i = 0; i < 3; i++) {
        js.executeScript("window.scrollBy(0, 500)");
        sleep(500);
    }
    // Scroll back to top
    js.executeScript("window.scrollTo(0, 0)");
} catch (Exception e) {
    System.out.println("Error while scrolling: " + e.getMessage());
}


ivate static void scrollToElement(WebDriver driver, WebElement element) {
```

```java
private static void scrollToElement(WebDriver driver, WebElement element) {
    try {
        JavascriptExecutor js = (JavascriptExecutor) driver;
        js.executeScript("arguments[0].scrollIntoView({block: 'center'});", element);
        sleep(500);
    } catch (Exception e) {
        System.out.println("Error scrolling to element: " + e.getMessage());
    }
}
private static WebElement getParentContainer(WebDriver driver, WebElement element) {
    try {
        WebElement container = null;
        try {
            container = element.findElement(By.xpath("./ancestor::div[contains(@class, 's-result-item') or cont
        } catch (NoSuchElementException e) {
            JavascriptExecutor js = (JavascriptExecutor) driver;
            container = (WebElement) js.executeScript(
                "function getProductContainer(el) {" +
                "  let current = el;" +
                "  for (let i = 0; i < 5; i++) {" +
                "    if (!current) return null;" +
                "    if (current.classList && " +
                "        (current.classList.contains('s-result-item') || " +
                "         current.classList.contains('sg-col'))) {" +
                "      return current;" +
                "    }" +
                "    current = current.parentElement;" +
                "  }" +
                "  return current; // Return the 5th ancestor even if no match" +
                "}" +
                "return getProductContainer(arguments[0]);",
                element);
        }

        return container;
    } catch (Exception e) {
        System.out.println("Error finding parent container: " + e.getMessage());
        return null;
```

```
Automation allowed only between 15:00 to 18:00. Current time: 11:00:23.757974800
```



```
Initializing Chrome WebDriver...
May 23, 2025 11:02:01 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find CDP implementation matching 136
May 23, 2025 11:02:01 AM org.openqa.selenium.chromium.ChromiumDriver lambda$new$5
WARNING: Unable to find version of CDP to use for 136.0.7103.114. You may need to include a
Navigating to Amazon...
Successfully loaded Amazon website
Looking for search box...
Submitting search query: laptop
Search results loaded for: laptop
Attempting to apply price filter (> ₹2000)
Standard price filter failed, trying alternative method...
Using JavaScript to apply price filter...
Price filter applied successfully
Attempting to apply rating filter (4 stars & up)
Using JavaScript to apply rating filter...
Rating filter applied successfully

Gathering product information...
Found 24 total products on page

Products starting with 'C' (Top 0):
--------------------------------------------------
No products found starting with 'C'. Showing other results instead:
--------------------------------------------------
1. HP 15, 13th Gen Intel Core i3-1315U, 8GB DDR4, 512GB SSD, (Win 11, ... ₹38,450
2. HP 15, 13th Gen Intel Core i5-1334U Laptop (16GB DDR4,512GB SSD) An... ₹38,450
3. ASUS TUF Gaming A15, AMD Ryzen 7 7435HS Gaming Laptop(NVIDIA RTX 30... ₹38,450
4. ASUS TUF Gaming A15, 15.6" (39.62cm) FHD 16:9 144Hz, AMD Ryzen 7 74... ₹38,450
5. Acer [SmartChoice Aspire 3 Laptop Intel Core Celeron N4500 Processo... ₹38,450

Automation complete.
```

Task 3:

```java
package com.gvp.com.HibernateCRUDApp2;

import org.openqa.selenium.By;

public class LoginProfileValidation {

    public static void main(String[] args) {

        Logger.getLogger("org.openqa.selenium").setLevel(Level.SEVERE);

        LocalTime now = LocalTime.now();
        LocalTime start = LocalTime.of(12, 0);
        LocalTime end = LocalTime.of(15, 0);

        if (now.isBefore(start) || now.isAfter(end)) {
            System.out.println("Test can only run between 12 PM and 3 PM. Current time: " + now);
            return;
        }

        WebDriver driver = null;
        Scanner scanner = new Scanner(System.in);

        try {

            ChromeOptions options = new ChromeOptions();
            options.addArguments("--start-maximized");
            options.addArguments("--disable-notifications");
            options.addArguments("--disable-blink-features=AutomationControlled");
            options.addArguments("--user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
            options.setExperimentalOption("excludeSwitches", java.util.Arrays.asList("enable-automation"));
            options.setExperimentalOption("useAutomationExtension", false);
            options.setCapability(CapabilityType.UNHANDLED_PROMPT_BEHAVIOUR, UnexpectedAlertBehaviour.DISMISS);


            System.setProperty("webdriver.chrome.silentOutput", "true");
            System.setProperty("webdriver.chrome.verboseLogging", "false");
```

```
System.out.println("Starting test - initializing browser...");


driver = new ChromeDriver(options);
driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));
WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));


((JavascriptExecutor) driver).executeScript("Object.defineProperty(navigator, 'webdriver', {get: () =>

System.out.println("📌 IMPORTANT INSTRUCTIONS 📌");
System.out.println("Due to Amazon's security measures, this test requires manual assistance.");
System.out.println("The browser will open to Amazon's homepage.");
System.out.println("Please follow these steps:");
System.out.println("1. Log in manually to your Amazon account");
System.out.println("2. Once logged in, navigate to your account page");
System.out.println("3. Return to this console and press Enter");
System.out.println("The test will then continue automatically to validate your profile.\n");


System.out.println("Step 1: Opening Amazon homepage...");
driver.get("https://www.amazon.in/");

System.out.println("\n Please log in manually and press Enter when ready...");
scanner.nextLine();

System.out.println("Continuing test...");
System.out.println("Current URL: " + driver.getCurrentUrl());
System.out.println("Current page title: " + driver.getTitle());


takeScreenshot(driver, "after_manual_login");


boolean isLoggedIn = false;
String profileName = "";
```

```
try {
    WebElement accountElement = wait.until(
        ExpectedConditions.visibilityOfElementLocated(By.id("nav-link-accountList"))
    );

    String accountText = accountElement.getText();
    System.out.println("Account element text: " + accountText);

    if (accountText.contains("Hello,") || accountText.contains("Sign Out")) {
        isLoggedIn = true;


        if (accountText.contains("Hello,")) {
            String[] parts = accountText.split("Hello,");
            if (parts.length > 1) {
                profileName = parts[1].trim().split("\\s+")[0].trim();
            }
        }
    }
} catch (Exception e) {
    System.out.println("Could not find account list element: " + e.getMessage());
}


if (!isLoggedIn || profileName.isEmpty()) {
    try {
        WebElement nameElement = driver.findElement(By.id("nav-link-accountList-nav-line-1"));
        String nameText = nameElement.getText();
        System.out.println("Nav line text: " + nameText);

        if (nameText.contains("Hello,")) {
            isLoggedIn = true;
            profileName = nameText.replace("Hello,", "").trim();
        }
    } catch (Exception e) {
        System.out.println("Could not find nav-line-1 element: " + e.getMessage());
```

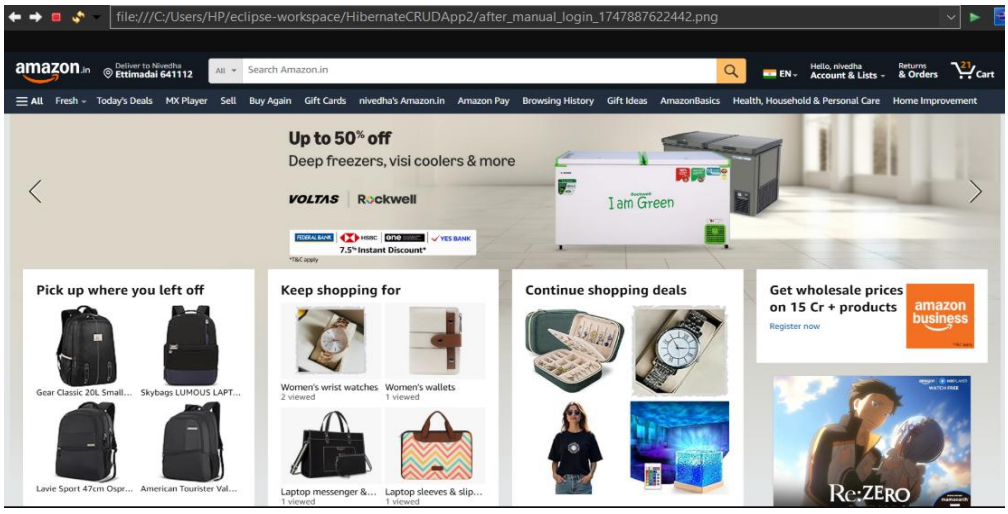
```
if (!isLoggedIn || profileName.isEmpty()) {
    try {

        WebElement accountMenu = driver.findElement(By.id("nav-link-accountList"));
        accountMenu.click();

        Thread.sleep(1000);
        WebElement accountInfo = driver.findElement(By.cssSelector(".nav-panel .nav-panel-name"));
        String accountInfoText = accountInfo.getText();
        System.out.println("Account info text: " + accountInfoText);

        isLoggedIn = true;
        profileName = accountInfoText.trim();

    } catch (Exception e) {
        System.out.println("Could not find account info in dropdown: " + e.getMessage());
    }
}

if (isLoggedIn) {
    System.out.println("\n☑ User is logged in successfully!");

    if (!profileName.isEmpty()) {
        System.out.println("Profile Name: " + profileName);

        Pattern forbiddenPattern = Pattern.compile("[ACGILK]");
        if (forbiddenPattern.matcher(profileName).find()) {
            System.out.println(" Validation FAILED: Profile name contains forbidden characters (A, C, G
        } else {
            System.out.println(" Validation PASSED: Profile name is valid (no forbidden characters).");
        }
    } else {
        System.out.println("Could not extract profile name, but user appears to be logged in.");
    }
} else {
    System.out.println("\nCould not verify if user is logged in.");
```

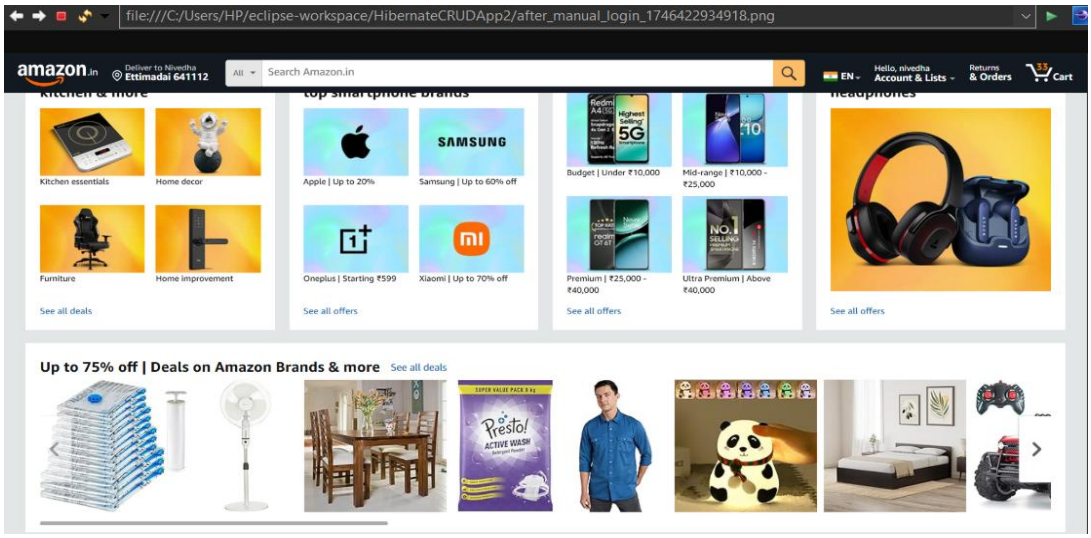
```
            System.out.println("Please ensure you completed the login process before pressing Enter.");
        }

    } catch (Exception e) {
        System.out.println("\n An unexpected error occurred:");
        e.printStackTrace();


        if (driver != null) {
            takeScreenshot(driver, "error_screenshot");
        }
    } finally {
        if (scanner != null) {
            scanner.close();
        }

        if (driver != null) {
            System.out.println("\nClosing browser...");
            driver.quit();
        }

        System.out.println("\n Test completed.");
    }
}

private static void takeScreenshot(WebDriver driver, String fileNamePrefix) {
    try {
        byte[] screenshotBytes = ((org.openqa.selenium.TakesScreenshot) driver).getScreenshotAs(org.openqa.sele
        Path path = Paths.get(fileNamePrefix + "_" + System.currentTimeMillis() + ".png");
        Files.write(path, screenshotBytes);
        System.out.println(" Screenshot saved to: " + path.toAbsolutePath());
    } catch (Exception e) {
        System.out.println("Failed to save screenshot: " + e.getMessage());
    }
}
```

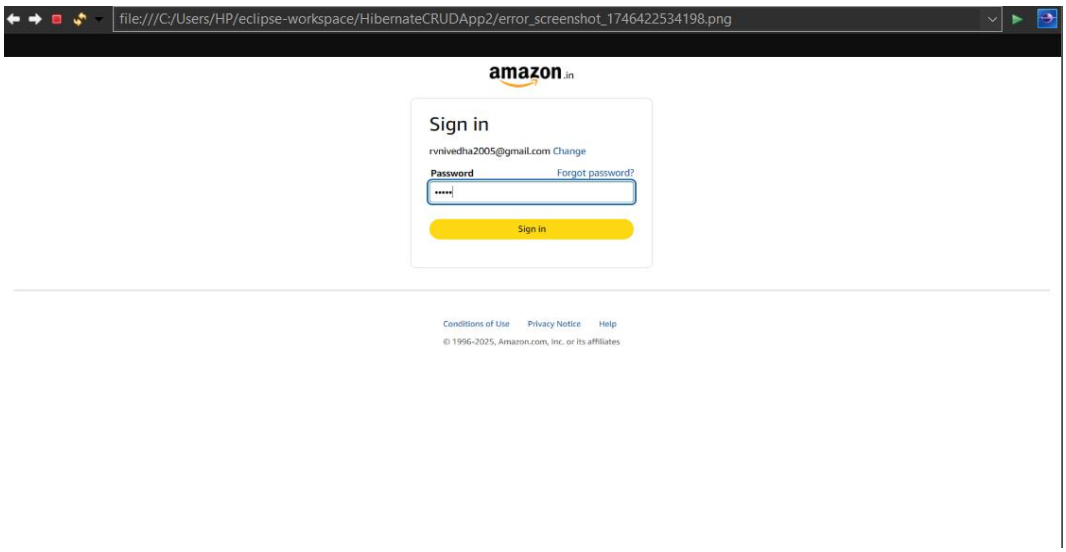
```
Test can only run between 12 PM and 3 PM. Current time: 11:12:01.600266900
```

```
Starting test - initializing browser...
📌 IMPORTANT INSTRUCTIONS 📌
Due to Amazon's security measures, this test requires manual assistance.
The browser will open to Amazon's homepage.
Please follow these steps:
1. Log in manually to your Amazon account
2. Once logged in, navigate to your account page
3. Return to this console and press Enter
The test will then continue automatically to validate your profile.

Step 1: Opening Amazon homepage...

 Please log in manually and press Enter when ready...

Continuing test...
Current URL: https://www.amazon.in/?ref_=nav_signin
Current page title: Online Shopping site in India: Shop Online for Mobiles, Books, Watches, Shoes and More - Amazon.in
 Screenshot saved to: C:\Users\HP\eclipse-workspace\HibernateCRUDApp2\after_manual_login_1747979140417.png
Account element text: Hello, nivedha
Account & Lists

☑ User is logged in successfully!
Profile Name: nivedha
 Validation PASSED: Profile name is valid (no forbidden characters).

Closing browser...

 Test completed.
```

Screenshot after manual login




file:///C:/Users/HP/eclipse-workspace/HibernateCRUDApp2/after_manual_login_1746422934918.png
amazon.in
Deliver to Nivedha
Ettimadai 641112
Search Amazon.in
EN
Hello, nivedha
Account & Lists
Returns
& Orders
Cart
Kitchen essentials
Home decor
Furniture
Home improvement
See all deals
Apple | Up to 20%
Samsung | Up to 60% off
Oneplus | Starting ₹599
Xiaomi | Up to 70% off
See all offers
Budget | Under ₹10,000
Mid-range | ₹10,000 - ₹25,000
Premium | ₹25,000 - ₹40,000
Ultra Premium | Above ₹40,000
See all offers
See all offers
Up to 75% off | Deals on Amazon Brands & more
See all deals


file:///C:/Users/HP/eclipse-workspace/HibernateCRUDApp2/error_screenshot_1746422534198.png
amazon.in
Sign in
rvnivedha2005@gmail.com Change
Password
Forgot password?
Sign in
Conditions of Use
Privacy Notice
Help
© 1996-2025, Amazon.com, Inc. or its affiliates

CHALLENGES AND SOLUTION:

- Time restricted execution: Used LocalTime.now() and conditional logic for hour checking
- Changing UI elements: Introduced fallback selectors
- Profile name extraction failures: Handled with try catch blocks.
- Anti scraping mechanism: Added chromeOptions to reduce detection.

OUTCOMES AND IMPACT

- Built scalable test automation scripts using industry standards.
- Understood how to test real-world web applications responsibly.
- Gained hands-on experience with Selenium and testing frameworks under guided mentorship.

CONCLUSION:

This internship at Null Class provided valuable practical exposure to automation testing and real-world use cases. I gained confidence in using Selenium WebDriver and applying robust coding patterns for building maintainable test scripts.