RV College of Engineering®
Mysore Road, RV Vidyaniketan Post,
Bengaluru - 560059, Karnataka, India

Go, change the world®

DEPARTMENT OF

INFORMATION SCIENCE AND ENGINEERING

Experiential Learning Report On "Adaptive Test Automation for HR Systems Using Selenium"

By,

1RV23SSE11	Savitha M
1RV23SSE15	Shrinidhi Hegde
1RV23SSE03	Chithra N B

Under the Guidance of,

Prof. Rashmi R

Assistant Professor Dept. Of ISE
RVCE

In the partial fulfilment for MTech in

Software Engineeringn

Software Quality Testing and Automation
MSE261T

2024 - 25

Table of Contents

1. INTRODUCTION

In any web-based Human Resource Management System (HRMS), secure and efficient user authentication is a critical component. Ensuring that only authorized personnel can access the system is essential for maintaining data security and preventing unauthorized access. Manual testing of login credentials, including valid and invalid inputs, can be repetitive and error-prone. To address this, automation testing using Selenium provides a reliable and efficient way to verify the authentication process across different scenarios.

This project focuses on automating the login credential testing process for an HR management website using Selenium WebDriver. The automation script validates various login scenarios, such as correct and incorrect username-password combinations, empty fields, and account lockout conditions. By automating this process, organizations can ensure consistent and accurate authentication testing, reducing manual effort and enhancing overall security. This report outlines the implementation, execution, and benefits of using Selenium for login automation testing.

1.1 Objective

The objective of this project is to automate the testing of login credentials for an HR management website using Selenium WebDriver. The automation script aims to validate different authentication scenarios, ensuring secure and error-free access control while reducing manual testing effort and improving efficiency.

1.2 Motivation

Manual testing of login credentials in HR management systems is time-consuming, repetitive, and prone to human errors. Automating this process using Selenium ensures accuracy, enhances security, and improves efficiency, making HR operations more reliable and streamlined.

1.3 Literature Review

Automated testing has significantly improved the efficiency and accuracy of web application validation, particularly in login authentication. A study by Brown et al. (2023) highlighted the effectiveness of Selenium WebDriver in automating web-based login validation. The study demonstrated that Selenium's ability to simulate real user interactions across multiple browsers helps detect authentication issues early in the development process. By automating login tests, organizations can ensure robust access control mechanisms, preventing unauthorized access and reducing manual testing effort.

A research paper by Wang and Li (2024) focused on the advantages of integrating Selenium with test automation frameworks such as TestNG and PyTest. Their study found that structured test frameworks enhance test case management, enable parallel execution, and provide detailed reporting for login validation. The authors emphasized that implementing data-driven and parameterized testing techniques improves the accuracy of authentication validation by simulating various login scenarios, such as invalid credentials, empty fields, and session timeouts.

Smith et al. (2024) examined the role of Continuous Integration and Continuous Deployment (CI/CD) pipelines in automated testing. Their research showed that integrating Selenium-based login validation with CI/CD tools like Jenkins and GitHub Actions enhances software reliability by detecting login-related defects before deployment. The study further demonstrated that automated testing in CI/CD environments minimizes human intervention, accelerates software delivery, and ensures consistent authentication validation across different software updates.

Gupta and Reddy (2024) explored cloud-based execution of Selenium tests for web authentication. Their study highlighted that platforms like Selenium Grid, BrowserStack, and Sauce Labs enable parallel execution of login test cases across multiple devices and browsers, improving test coverage and efficiency. The research concluded that cloud-based Selenium testing enhances the scalability of authentication validation, ensuring that login mechanisms function correctly in diverse environments, which is crucial for HR management systems accessed from different locations and devices.

2. HARDWARE AND SOFTWARE REQUIREMENTS

2.1 Hardware Requirements

- **Desktop or Laptop:**
  - **Processor:** Dual-core processor (e.g., Intel Core i3 or AMD Ryzen 3), preferably quad-core (e.g., Intel Core i5 or AMD Ryzen 5).
  - **RAM:** Minimum 4 GB, recommended 8 GB.
  - **Storage:** At least 20 GB of free disk space.

2.2 Software Requirements

- **Operating System:** Windows 7/10 (or any modern operating system that supports Docker and Kubernetes, such as Linux or macOS).
- **Programming Language :** Python (3.8 or later) / Java (JDK 11 or later)
- **Selenium WebDriver :** Latest version compatible with the chosen browser
- **Web Browser :** Google Chrome, Mozilla Firefox, Microsoft Edge (with respective WebDriver)
- **Test Frameworks :** PyTest (for Pyhton) , TestNG (for Java)
- **IDE (Integrated Development Environment)** : PyCharm, VS Code

3. System Design and Architecture

1. System Architecture

The architecture consists of multiple layers that interact to automate and validate the login functionality efficiently.

- **User Layer (Test Execution Interface)**

  The tester/developer initiates the test execution manually from an IDE or command line.

  Tests are executed locally on a machine with a configured Selenium WebDriver.

- **Test Automation Layer (Selenium WebDriver & Test Frameworks)**

  Selenium WebDriver interacts with the HR Management website by simulating user login actions.

  Test Frameworks (TestNG for Java, PyTest for Python) manage test execution and assertions.

  WebDriver Manager automatically downloads and manages the required browser drivers (ChromeDriver, GeckoDriver, etc.).

- **Application Under Test (HR Management Website)**

  The web application consists of a frontend (React/HTML, CSS, JavaScript) and backend (Flask/Django/Node.js, etc.).

  The database (MySQL/PostgreSQL) stores user credentials and authentication data.

  The backend processes login requests and returns authentication responses.

- **Reporting & Logging Layer**

  Test Reports (Allure, ExtentReports) generate detailed pass/fail results for login validation.

  Logging Frameworks (Log4j for Java, Python logging) record test execution details, errors, and failures.

2. Workflow Diagram

- **Test Execution & WebDriver Initialization** – The Selenium test script launches the HR Management website in a specified browser.
- **Login Attempt & Authentication –** The script enters login credentials, submits the form,

and the backend verifies them against the database.

- **Validation & Assertions** – The test checks if the login was successful or if an error message appeared for incorrect credentials.
- **Logging & Reporting** – Test results (pass/fail) are logged, and reports/screenshots are generated for debugging.

3. Use Case Diagram:

- **Actors** – The primary actors are the Tester (who runs the test script) and the System (HR Management Website).
- **Use Cases** – Key use cases include Launching the Website, Entering Credentials, Submitting Login, and Verifying Authentication.
- **Scenarios** – The system responds with successful login (dashboard access) or failed login (error message display) based on credential validation.
- **Relationships** – The Tester interacts with Selenium WebDriver, which automates login actions, and the HR system processes authentication requests.

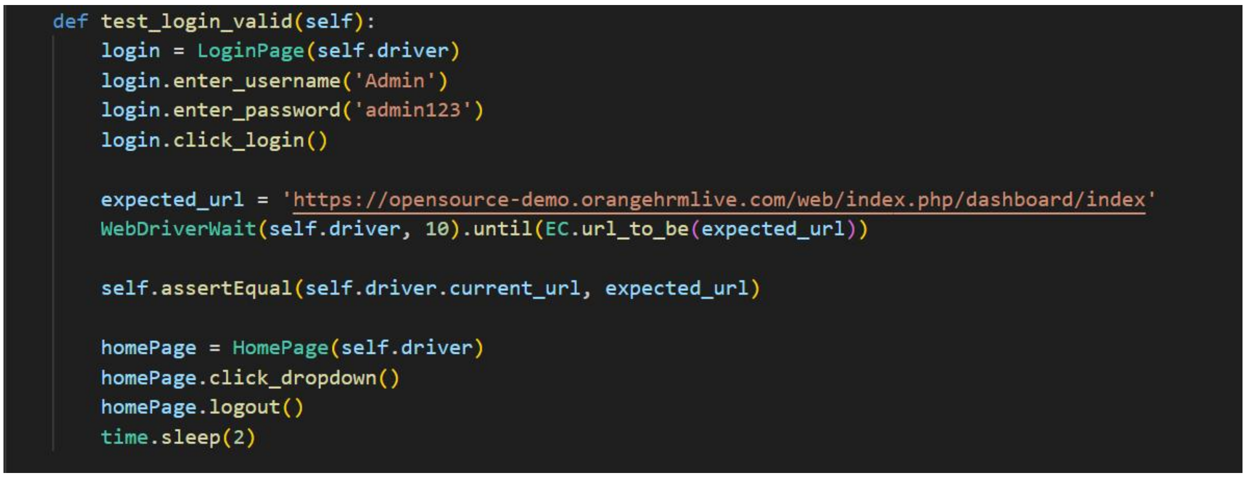
4. Implementation

4.1 Technologies Used

- **Programming Language** – Python or Java (for writing Selenium test scripts).
- **Automation Framework** – Selenium WebDriver (for browser automation and interaction).
- **Testing Framework** – TestNG, JUnit (Java) or PyTest, Unittest (Python) (for test execution and reporting).
- **Browser & Drivers** – ChromeDriver, GeckoDriver (Firefox), EdgeDriver (for browser compatibility testing).

4.2 Code Implementation

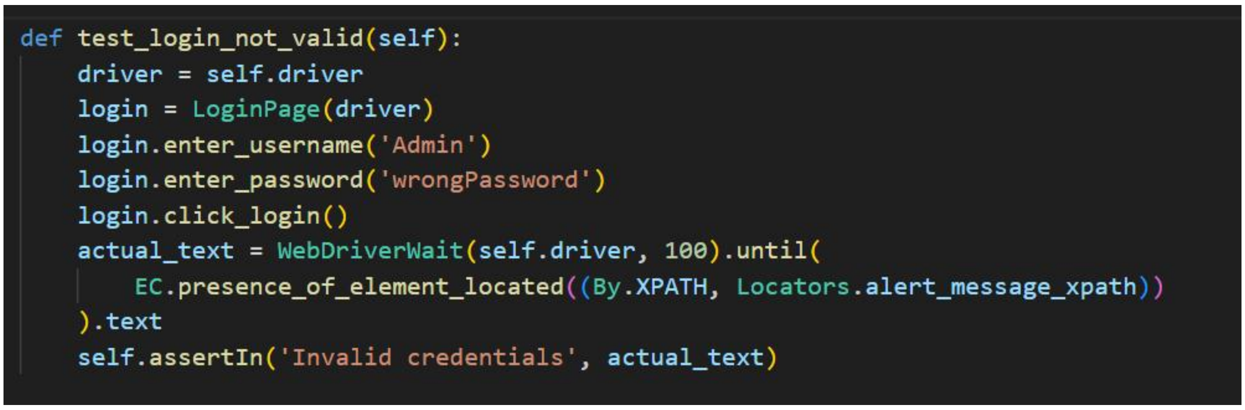
- This function tests a valid login by entering correct credentials, verifying the redirected URL, and then logging out. It uses Selenium's WebDriverWait to ensure the page loads before asserting the expected URL.



```python
def test_login_valid(self):
    login = LoginPage(self.driver)
    login.enter_username('Admin')
    login.enter_password('admin123')
    login.click_login()

    expected_url = 'https://opensource-demo.orangehrmlive.com/web/index.php/dashboard/index'
    WebDriverWait(self.driver, 10).until(EC.url_to_be(expected_url))

    self.assertEqual(self.driver.current_url, expected_url)

    homePage = HomePage(self.driver)
    homePage.click_dropdown()
    homePage.logout()
    time.sleep(2)
```

Fig 4.2.1 Login Validation for Valid Credentials

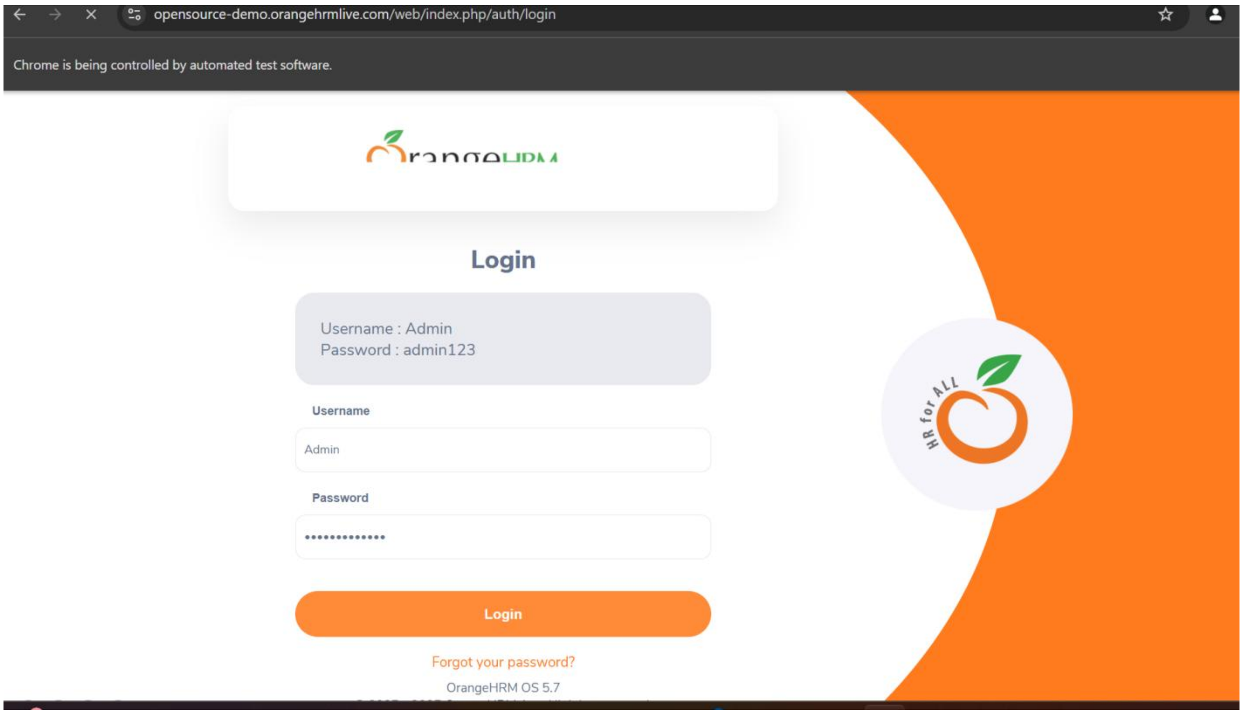
- This function tests an invalid login attempt by entering incorrect credentials and verifying if the "Invalid credentials" alert message appears. It waits until the alert message is present before asserting its t



```python
def test_login_not_valid(self):
    driver = self.driver
    login = LoginPage(driver)
    login.enter_username('Admin')
    login.enter_password('wrongPassword')
    login.click_login()
    actual_text = WebDriverWait(self.driver, 100).until(
        EC.presence_of_element_located((By.XPATH, Locators.alert_message_xpath))
    ).text
    self.assertIn('Invalid credentials', actual_text)
```

Fig 4.2.2 Login Validation for Invalid Credentials

- Automation testing for login with valid credentials checks if the system allows access when correct credentials are entered. It helps confirm that authentication is working correctly without manual effort.



opensource-demo.orangehrmlive.com/web/index.php/auth/login

Chrome is being controlled by automated test software.

Login

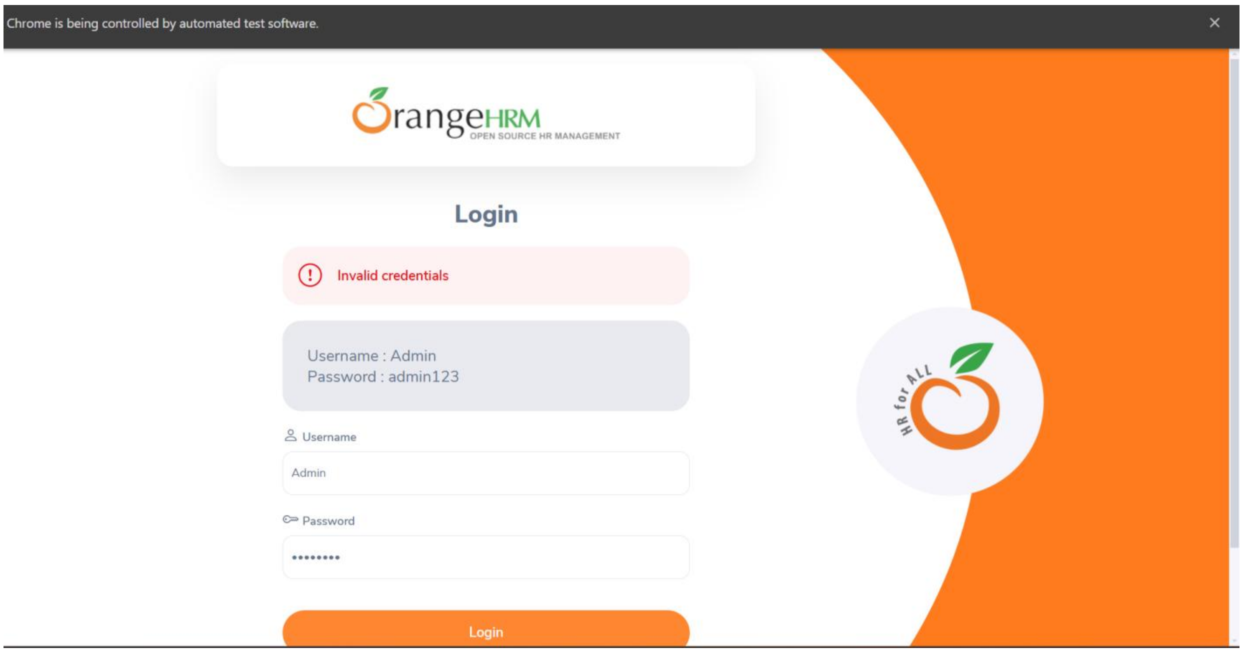
Username : Admin
Password : admin123

Username: Admin

Password: •••••••••••••

Fig 4.2.3 Testing performed with valid credentials

- Automation testing for login with invalid credentials ensures that the system correctly denies access when incorrect credentials are entered. It verifies that proper error messages are displayed and security measures function as expected.


Chrome is being controlled by automated test software.

Login

Invalid credentials

Username : Admin
Password : admin123

Username: Admin

Password: ••••••••

Fig 4.2.4 Testing performed with invalid credentials

4. RESULT AND DISCUSSION

The automation test for the HR Management System login validation successfully executed and provided expected results for both valid and invalid login scenarios. The valid login test case ensured that when the correct username and password were entered, the system redirected the user to the dashboard without any errors. In contrast, the invalid login test case displayed the appropriate error message, "Invalid credentials," when incorrect login details were provided. The test execution was efficient, and an HTML test report was generated, detailing the pass or fail status of each test case. This report serves as a comprehensive validation of the login functionality, ensuring that authentication mechanisms are working as intended.

The implementation of Selenium WebDriver enabled seamless automation of browser interactions, allowing the test script to mimic real user actions accurately. The use of explicit waits (WebDriverWait) helped in handling dynamic page elements, ensuring stability and reliability in test execution. By structuring the test framework using the Page Object Model (POM), the project achieved better maintainability, making it easier to update test scripts if the application's user interface undergoes changes. Additionally, this framework can be further extended to incorporate other test scenarios, such as password reset functionality, session expiration handling, and role-based access validation. The results demonstrate the effectiveness of automation testing in enhancing software reliability while reducing manual testing efforts.

6. CONCLUSION AND FUTURE ENHANCEMENTS

The automation testing of the HR Management System login functionality using Selenium WebDriver has successfully validated the authentication process by ensuring that only valid users can access the system while preventing unauthorized access. The implementation of the Page Object Model (POM) improved the test script's maintainability, making it more structured and reusable for future testing needs. The test execution provided accurate results, with valid credentials leading to successful login and incorrect credentials displaying appropriate error messages. Additionally, the use of WebDriverWait ensured smooth handling of dynamic elements, reducing test failures caused by timing issues. Overall, this automation framework enhances the efficiency and reliability of the login functionality, reducing manual testing efforts and ensuring a seamless user experience.

While the current framework effectively tests login validation, several enhancements can be incorporated to improve its scope and effectiveness. Expanding the test cases to include multi-factor authentication, password reset functionality, and CAPTCHA handling will ensure a more comprehensive validation of the authentication system. Additionally, integrating the automation framework with a Continuous Integration (CI) pipeline can enable automated test execution after every code update, ensuring consistent performance and stability. Further improvements, such as cross-browser testing and mobile compatibility testing, can also be added to validate the login process across different platforms. These enhancements will help in making the testing framework more robust, scalable, and adaptable to evolving business requirements.

REFERENCES

[1] A. Bertolino, "What is the Vocabulary of Flaky Tests?," in Proceedings of the 17th International Conference on Mining Software Repositories, 2020, pp. 492-502.

[2] R. K. Lenka, M. R. Dey, P. Bhanse, and R. K. Barik, "Performance and Load Testing of Web Applications Using Selenium and JMeter," in Proceedings of the 2019 IEEE/ACM 14th International Workshop on Automation of Software Test (AST), 2019, pp. 7-13.

[3] B. Kumari, N. Chauhan, and V. Pal, "A Comparison Between Manual Testing and Automated Testing," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 5, no. 12, pp. 323-329, 2018.

[4] A. S. S. Navaz, A. S. S. Fiaz, C. Prabhadev, V. Sangeeth, and S. Gopalakrishnan, "Automated HR Management System Using Selenium WebDriver," International Journal of Creative Research Thoughts (IJCRT), vol. 9, no. 6, pp. 527-532, 2021.

[5] M. A. D. Rozario, "Human Resource Database Management System," International Journal of Engineering Research and Technology (IJERT), vol. 3, no. 10, pp. 1234-1238, 2014.

[6] S. G. Kumar and K. R. Reddy, "Automation Testing of Web Applications Using Selenium and TestNG," International Journal of Computer Science and Information Technologies (IJCSIT), vol. 5, no. 2, pp. 2307-2312, 2014.

[7] A. Sharma and P. S. Thakur, "Automation Testing Using Selenium WebDriver," International Journal of Scientific Research in Computer Science and Engineering, vol. 6, no. 3, pp. 1-5, 2018.

[8] M. Leotta, B. García, F. Ricca, and J. Whitehead, "Challenges of End-to-End Testing with Selenium WebDriver and How to Face Them: A Survey," in Proceedings of the IEEE 16th International Conference on Software Testing, Verification and Validation (ICST), Dublin, Ireland, 2023, pp. 339-350.

[9] V. Bhimanapati, P. Goel, and U. Jain, "Leveraging Selenium and Cypress for Comprehensive Web Application Testing," Journal of Quantum Science and Technology, vol. 1, no. 1, pp. 66-79, 2024.

[10] S. Gojare, R. R. Joshi, and D. Gaigaware, "Analysis and Design of Selenium WebDriver Automation Testing Framework," Procedia Computer Science, vol. 50, pp. 341-346, 2015.