Document: Approach, Assumptions, and Instructions

1. Approach

Objective

To automate the login process of the OpenMRS platform using Selenium with Python and generate a test execution report. The project involves setting up an automation framework, automating the login process, verifying successful login, and handling errors.

Steps Taken

Setup Automation Framework:

Framework Chosen: Selenium with Python.

Environment Setup:

Installed Selenium using pip.

Set up Chrome WebDriver for browser automation.

Created a Python virtual environment for dependency management.

Automate the Login Process:

Login Test Script (login_test.py):

Opened the OpenMRS login page.

Entered valid credentials and attempted to log in.

Successful Login Test Script (successfullogin_test.py):

Verified that a specific element on the dashboard indicates a successful login.

Error Handling Test Script (errorhandling_test.py):

Tested incorrect login credentials.

Checked for the display of the error message for invalid credentials.

Verify Successful Login:

Added verification in the script to ensure that a specific element on the landing page confirms a successful login.

Error Handling:

Implemented error handling to check for the presence of error messages for invalid login attempts.

Captured screenshots and page sources for debugging if errors occur.

Test Execution Script (main_script.py):

Created a main script to run all individual test scripts and handle their execution status.

Captured and printed the output and errors of each script.

2. Assumptions

Test Environment: The OpenMRS platform is accessible at https://qa.kenyahmis.org/openmrs/spa/login and behaves consistently.

Credentials: Valid credentials are used for successful login tests, and incorrect credentials are used for error handling tests.

Element Identifiers: CSS selectors and IDs used in the test scripts accurately reflect the elements on the OpenMRS login page.

Python and Selenium: The Python environment has Selenium installed, and the appropriate WebDriver (e.g., ChromeDriver) is set up.

3. Instructions to Run the Script

Setup Environment:

Ensure Python is installed on your machine.

Install the required packages using pip:

bash

pip install selenium

Download and Setup WebDriver:

Download the appropriate WebDriver for Chrome browser (e.g., ChromeDriver) should be the corresponding version to that of the chrome browser and ensure it is in your system's PATH.

Place Scripts:

Ensure all test scripts (login_test.py, successfullogin_test.py, errorhandling_test.py, main_script.py) are in the same directory.

Run the Main Script:

Execute the main_script.py to run all test scripts sequentially:

bash

python main_script.py

Review Output:

Check the console output for the status of each test script.

Review any screenshots or page sources saved in case of errors for debugging.