Setup & ERP Automation

Complete Step-by-Step Robot Framework Setup & ERP Automation Guide

---

Installation tutorial videos:
https://youtu.be/2-s64hTj2IM?si=zyHx7gctqfTW-CyH

https://youtu.be/dIsrQ-FNu08?si=_Ptet03beAjntzdn

Step 1: Install Python

1. Download Python 3.10+ from:
   https://www.python.org/downloads/

2. Run the installer → **Check "Add Python to PATH"**
3. Verify installation:

```
python --version
```

- Should show: `Python 3.x.x`

4. Verify pip (Python package manager) works:

```
pip --version
```

Step 2: Install Robot Framework

1. Open Command Prompt.
2. Install Robot Framework:

```
pip install robotframework
```

3. Verify installation:

```
robot --version
```

Step 3: Install SeleniumLibrary

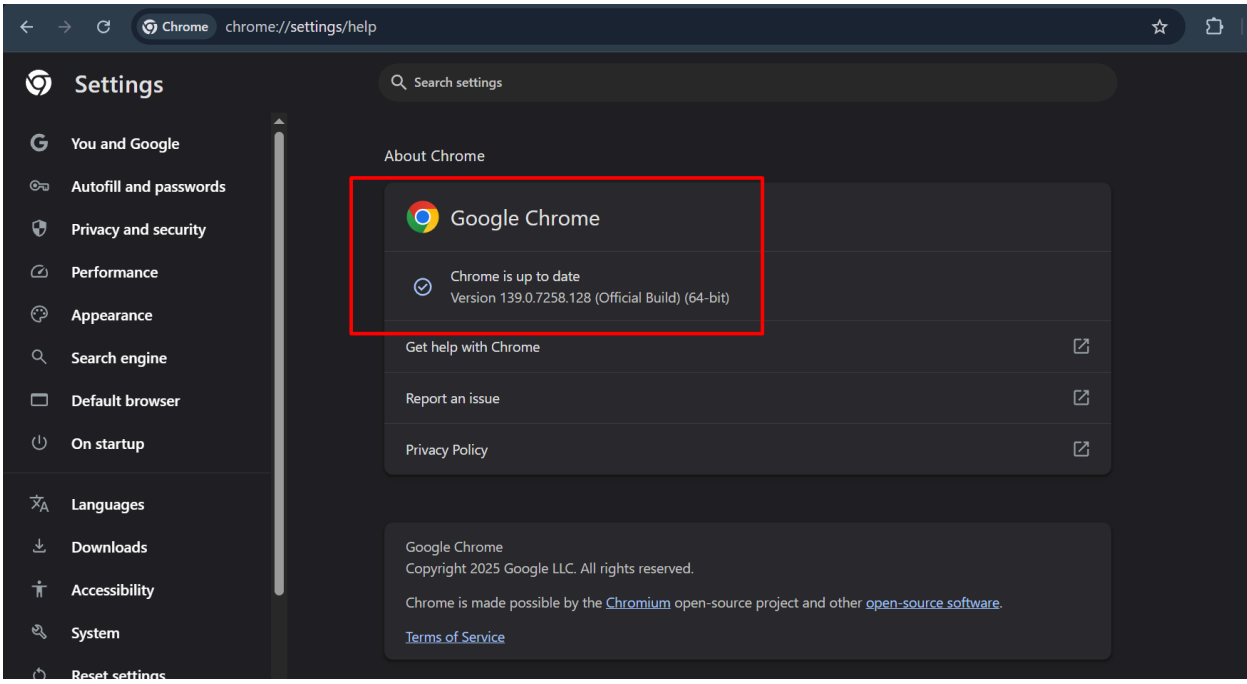
1. Install the Selenium library for Robot Framework:

```
pip install robotframework-seleniumlibrary
```

Step 4: Download & Setup ChromeDriver

1. Check your Chrome version:

    - Open Chrome → Go to `chrome://settings/help`



    - Example: `Version 139.0.7258.128`

2. Download matching ChromeDriver:
   https://googlechromelabs.github.io/chrome-for-testing/#stable

   - Select version **139.x**

   - Download for **Windows → chromedriver-win64.zip**

Binary	Platform	URL	HTTP status
chrome	linux64	https://storage.googleapis.com/chrome-for-testing-public/139.0.7258.68/linux64/chrome-linux64.zip	200
chrome	mac-arm64	https://storage.googleapis.com/chrome-for-testing-public/139.0.7258.68/mac-arm64/chrome-mac-arm64.zip	200
chrome	mac-x64	https://storage.googleapis.com/chrome-for-testing-public/139.0.7258.68/mac-x64/chrome-mac-x64.zip	200
chrome	win32	https://storage.googleapis.com/chrome-for-testing-public/139.0.7258.68/win32/chrome-win32.zip	200
chrome	win64	https://storage.googleapis.com/chrome-for-testing-public/139.0.7258.68/win64/chrome-win64.zip	200
chromedriver	linux64	https://storage.googleapis.com/chrome-for-testing-public/139.0.7258.68/linux64/chromedriver-linux64.zip	200
chromedriver	mac-arm64	https://storage.googleapis.com/chrome-for-testing-public/139.0.7258.68/mac-arm64/chromedriver-mac-arm64.zip	200
chromedriver	mac-x64	https://storage.googleapis.com/chrome-for-testing-public/139.0.7258.68/mac-x64/chromedriver-mac-x64.zip	200
chromedriver	win32	https://storage.googleapis.com/chrome-for-testing-public/139.0.7258.68/win32/chromedriver-win32.zip	200
chromedriver	win64	https://storage.googleapis.com/chrome-for-testing-public/139.0.7258.68/win64/chromedriver-win64.zip	200
chrome-headless-shell	linux64	https://storage.googleapis.com/chrome-for-testing-public/139.0.7258.68/linux64/chrome-headless-shell-linux64.zip	200
chrome-headless-shell	mac-arm64	https://storage.googleapis.com/chrome-for-testing-public/139.0.7258.68/mac-arm64/chrome-headless-shell-mac-arm64.zip	200
chrome-headless-shell	mac-x64	https://storage.googleapis.com/chrome-for-testing-public/139.0.7258.68/mac-x64/chrome-headless-shell-mac-x64.zip	200
chrome-headless-shell	win32	https://storage.googleapis.com/chrome-for-testing-public/139.0.7258.68/win32/chrome-headless-shell-win32.zip	200
chrome-headless-shell	win64	https://storage.googleapis.com/chrome-for-testing-public/139.0.7258.68/win64/chrome-headless-shell-win64.zip	200

3. Extract `chromedriver.exe` to a folder:

```
C:\WebDriver\bin     # create this folder
```

Step 5: Add ChromeDriver to System PATH

1. Press **Windows Key → search "Environment Variables" → Edit the system environment variables**

2. Click **Environment Variables → System variables → Path → Edit → New**

3. Add:

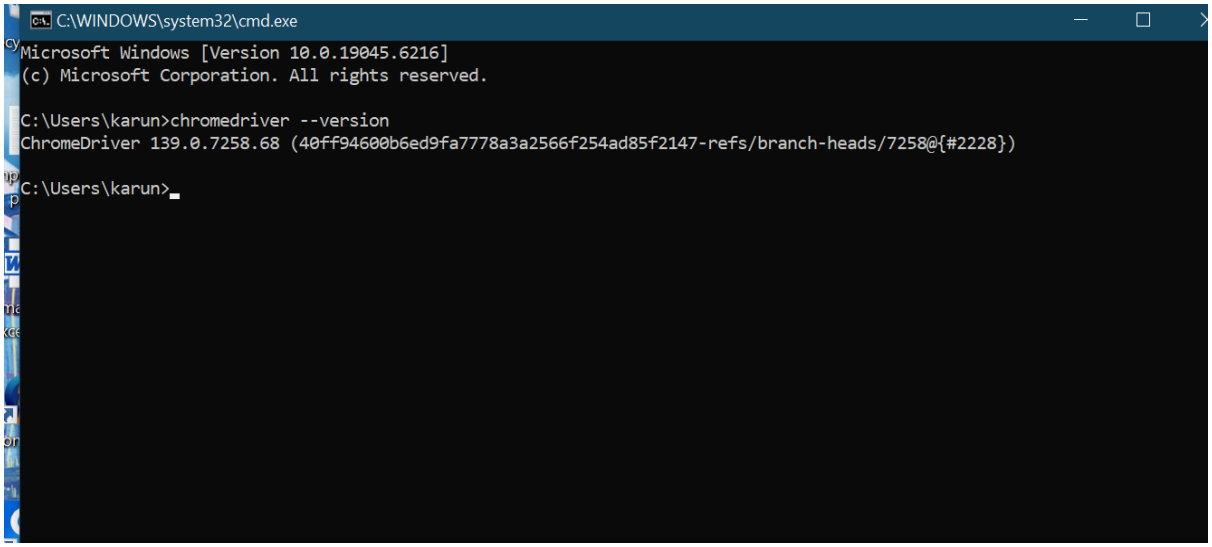
```
C:\WebDriver\bin
```

4. Click **OK**, save all changes.

5. Open a **new Command Prompt** and verify:

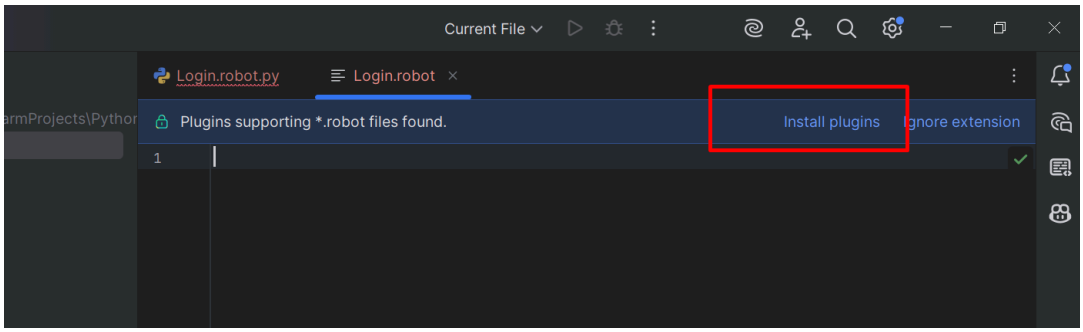
```
chromedriver --version
```

- Should show: `ChromeDriver 139.0.7258.xxx` ✅

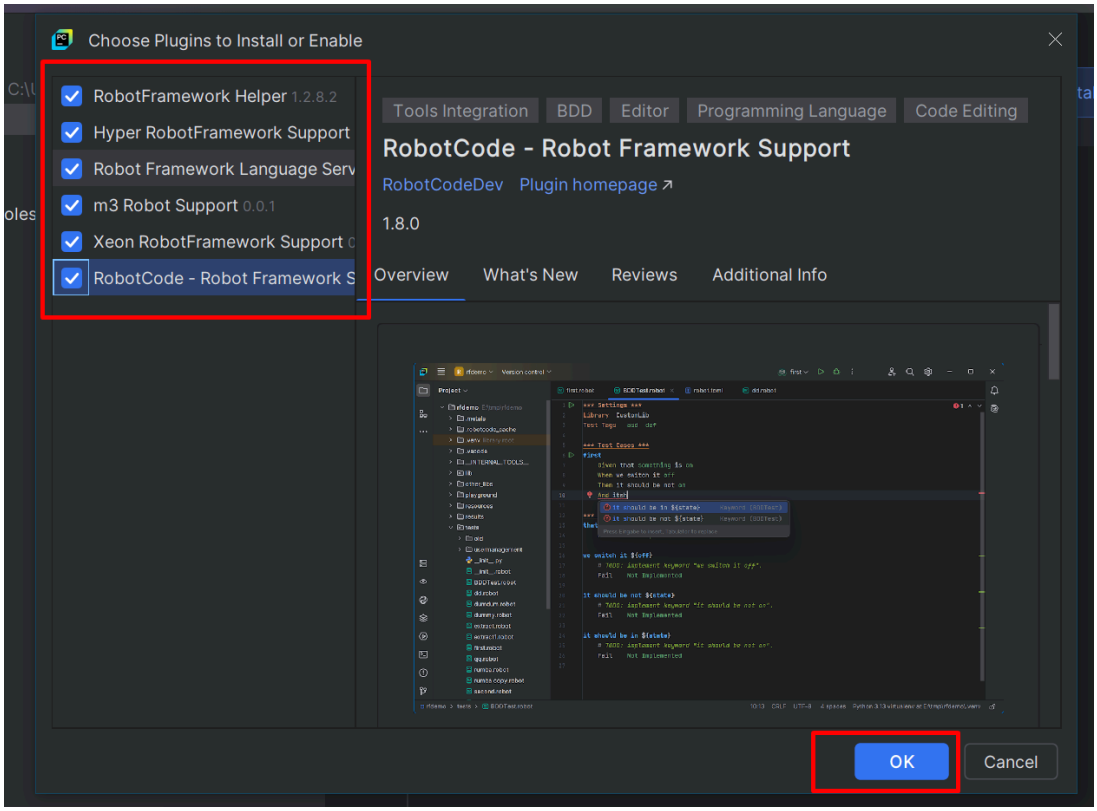


Step 6: Installing plugins in pycharm

1. In pycharm you can see this then click on install plugins



2.Select all check box and click on ok button

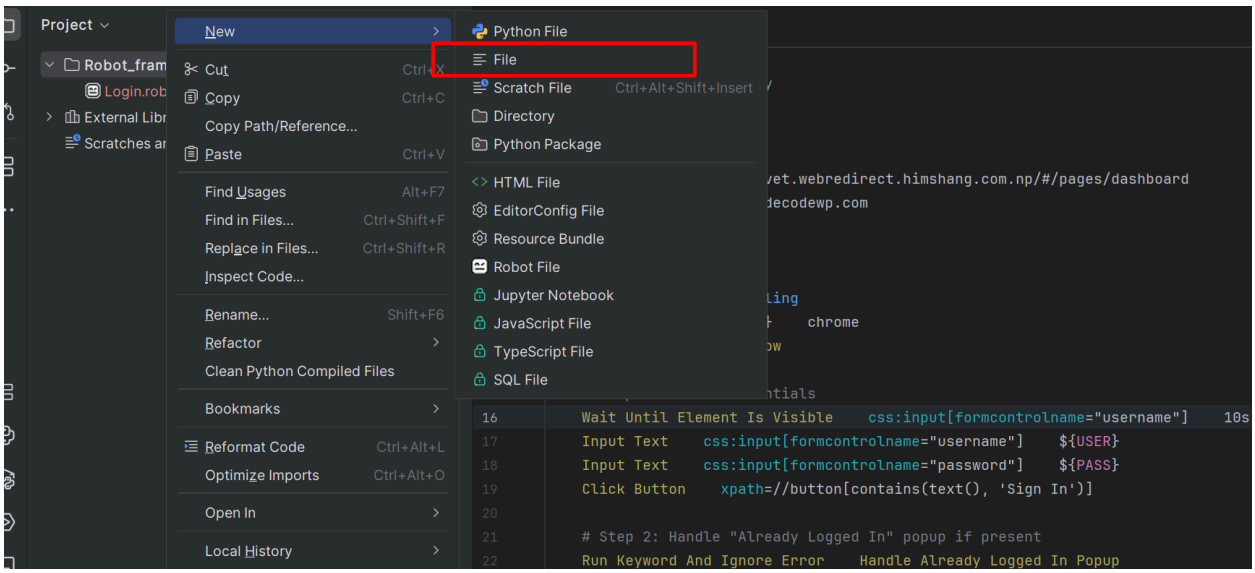


Then installation started of plugins

3. When installation completed restart pycharm

Step 7:Sample code run

1. Create file in pycharm with Extension (.robot)
   (Example: **login.robot**)



2. Take this sample for demo

```
*** Settings ***
Library     SeleniumLibrary
Library     Collections

*** Variables ***
${URL}
https://velvet.webredirect.himshang.com.np/#/pages/dashboar
d
${USER}         gedehim917@decodewp.com
${PASS}         Tebahal1!
```

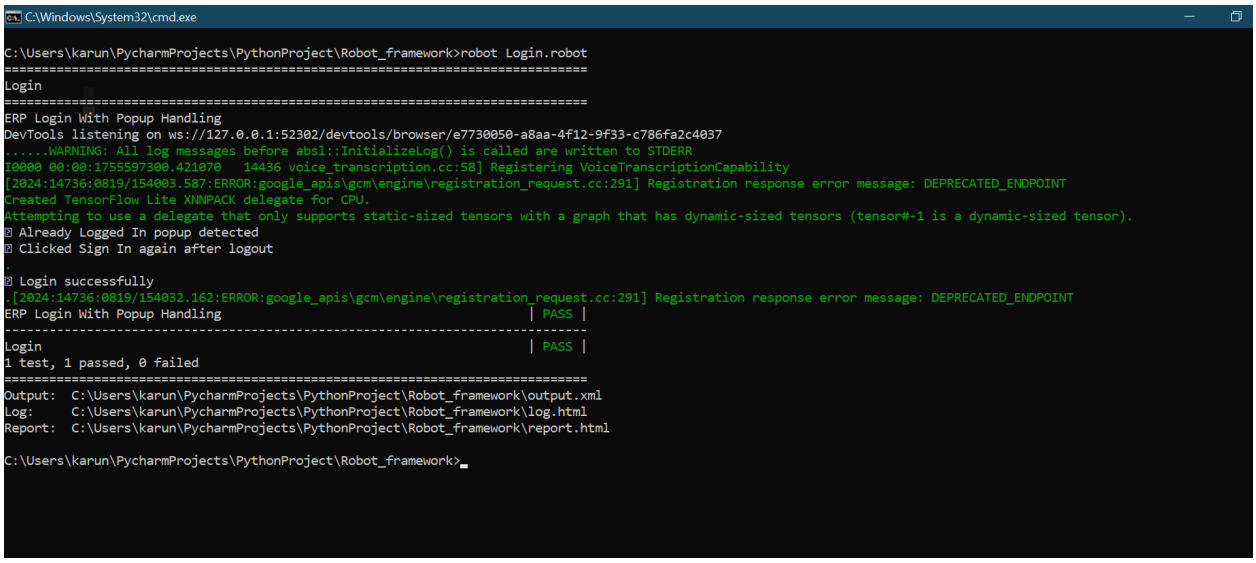
```
*** Test Cases ***
ERP Login With Popup Handling
    Open Browser    ${URL}    chrome
    Maximize Browser Window

    # Step 1: Enter credentials
    Wait Until Element Is Visible
css:input[formcontrolname="username"]    10s
    Input Text    css:input[formcontrolname="username"]
${USER}
    Input Text    css:input[formcontrolname="password"]
${PASS}
    Click Button    xpath=//button[contains(text(), 'Sign
In')]

    # Step 2: Handle "Already Logged In" popup if present
    Run Keyword And Ignore Error    Handle Already Logged In
Popup

    Log To Console    \n✓ Login successfully
    Sleep    30s
    Close Browser


*** Keywords ***
Handle Already Logged In Popup
    Wait Until Element Is Visible
xpath=//button[.//span[text()='Logout']]    20s
    Log To Console    ✓ Already Logged In popup detected
    Click Button    xpath=//button[.//span[text()='Logout']]
    Sleep    8s
    Wait Until Element Is Visible
xpath=//button[contains(text(), 'Sign In')]    10s
    Press Keys    xpath=//button[contains(text(), 'Sign
In')]    ENTER

    Log To Console    ✓ Clicked Sign In again after logout
```

3. Run the Test

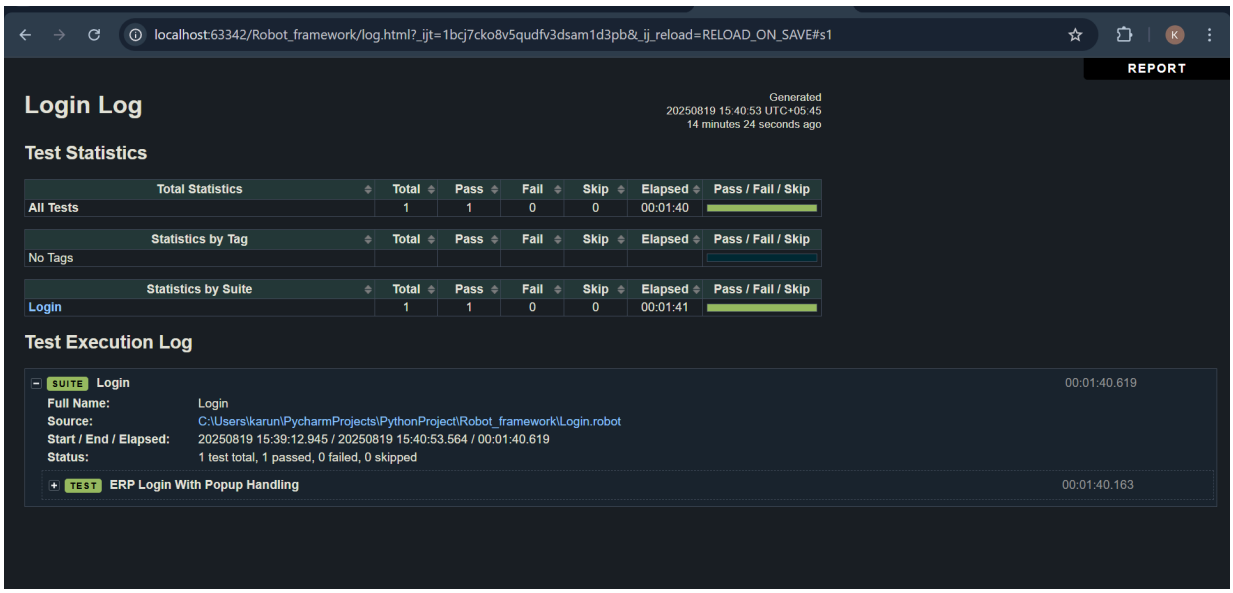
Open terminal and paste it :

```
robot login.robot
```

Result:



Report:



Important :

Press Keys keyword accepts unicode escape sequences.

For TAB, use \\09.

For ENTER, use \\13.

For example:

Press Keys locator \\09 → presses Tab

Press Keys locator \\13 → presses Enter