

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/dlsrQ-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

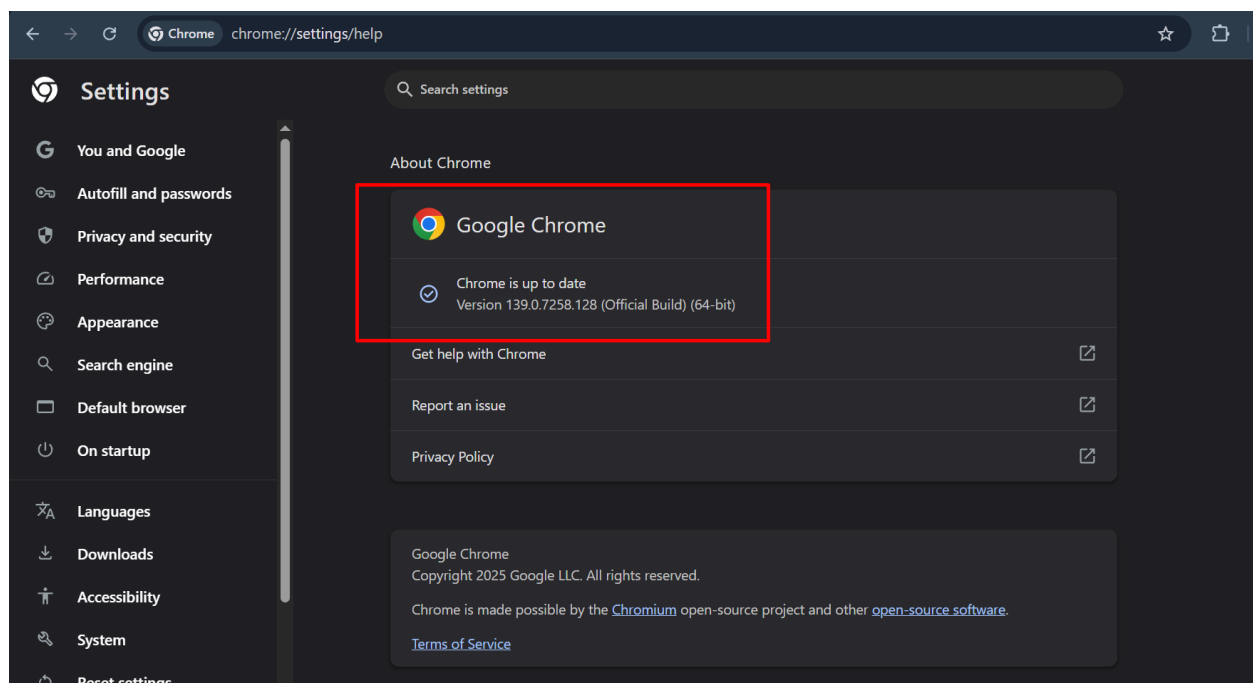
1. Install the Selenium library for Robot Framework:

```
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

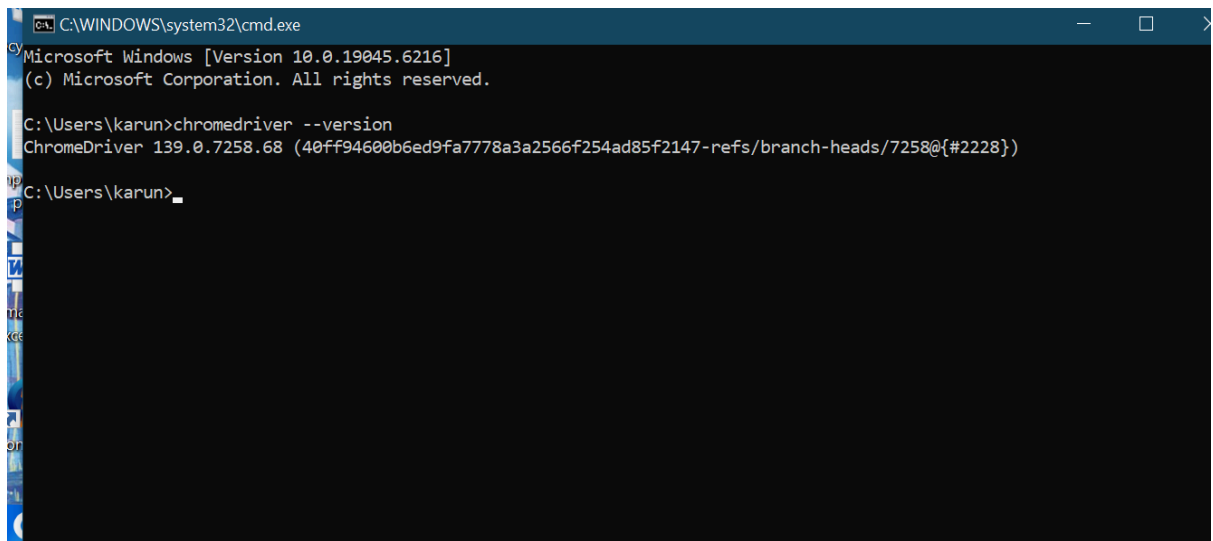
1. Press **Windows Key** → search “**Environment Variables**”
→ **Edit the system environment variables**
2. Click **Environment Variables** → **System variables** → **Path**
→ **Edit** → **New**
3. Add:

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** ✓



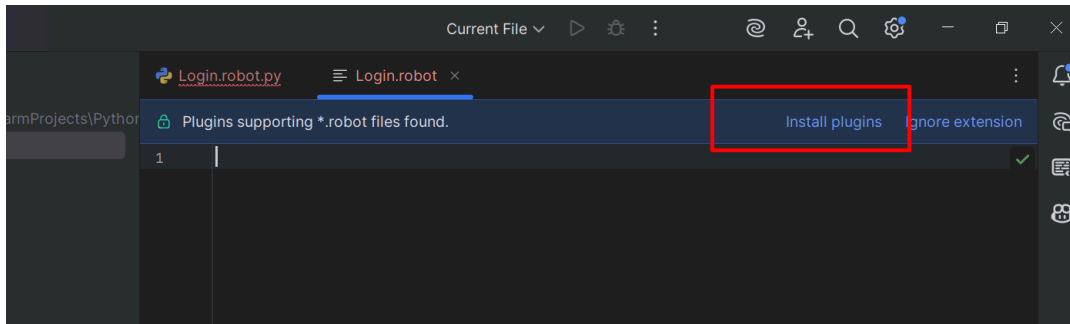
```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.6216]
(c) Microsoft Corporation. All rights reserved.

C:\Users\karun>chromedriver --version
ChromeDriver 139.0.7258.68 (40ff94600b6ed9fa7778a3a2566f254ad85f2147-refs/branch-heads/7258@{#2228})

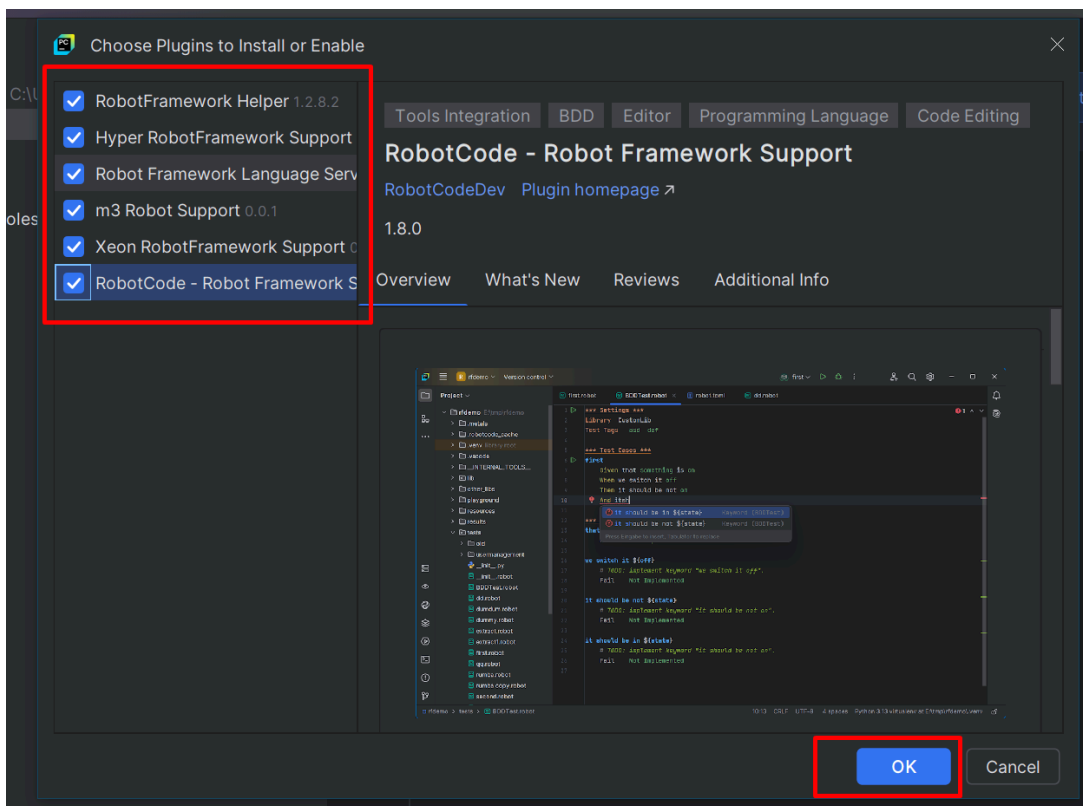
C:\Users\karun>
```

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

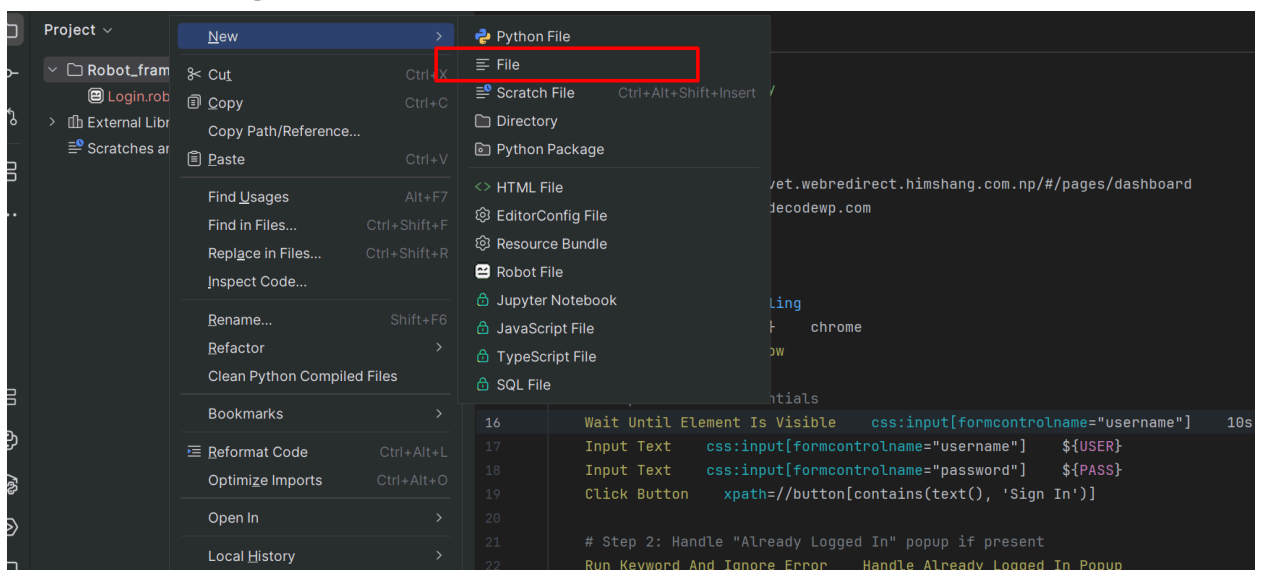


Then installation started of plugins

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/dashboard

${USER}      gedehim917@decodewp.com
${PASS}      Tebahall!
```

```

*** 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:

```
C:\Windows\System32\cmd.exe
C:\Users\karun\PycharmProjects\PythonProject\Robot_framework>robot Login.robot
=====
Login
=====
ERP Login With Popup Handling
DevTools listening on ws://127.0.0.1:52302/devtools/browser/e7730050-a8aa-4f12-9f33-c786fa2c4037
...-WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
10000 00:00:1755597300.421078 14436 voice transcription.cc:58] Registering VoiceTranscriptionCapability
[2024:14736:0819/154003.587:ERROR:google_apis\gcm\engine\registration_request.cc:291] Registration response error message: DEPRECATED_ENDPOINT
Created TensorFlow Lite XNNPACK delegate for CPU.
Attempting to use a delegate that only supports static-sized tensors with a graph that has dynamic-sized tensors (tensor#-1 is a dynamic-sized tensor).
[ Already logged in popup detected
[ Clicked Sign In again after logout

[ Login successfully
[2024:14736:0819/154032.162:ERROR:google_apis\gcm\engine\registration_request.cc:291] Registration response error message: DEPRECATED_ENDPOINT
ERP Login With Popup Handling | PASS |
-----
Login | PASS |
1 test, 1 passed, 0 failed
=====
Output: C:\Users\karun\PycharmProjects\PythonProject\Robot_framework\output.xml
Log: C:\Users\karun\PycharmProjects\PythonProject\Robot_framework\log.html
Report: C:\Users\karun\PycharmProjects\PythonProject\Robot_framework\report.html
C:\Users\karun\PycharmProjects\PythonProject\Robot_framework>
```

Report:

The screenshot shows a web browser displaying a test report. The report is titled "Login Log" and was generated on 20250819 15:40:53 UTC+05:45, 14 minutes and 24 seconds ago. It includes a "Test Statistics" section with a table showing 1 total test, 1 passed, 0 failed, and 0 skipped. The "Test Execution Log" section shows a suite named "Login" with a duration of 00:01:40.619, and a test case "ERP Login With Popup Handling" with a duration of 00:01:40.163.

Total Statistics	Total	Pass	Fall	Skip	Elapsed	Pass / Fall / Skip
All Tests	1	1	0	0	00:01:40	1 / 0 / 0

Statistics by Suite	Total	Pass	Fall	Skip	Elapsed	Pass / Fall / Skip
Login	1	1	0	0	00:01:41	1 / 0 / 0

Test Execution Log

- SUITE** Login (00:01:40.619)
 - Full Name: Login
 - Source: C:\Users\karun\PycharmProjects\PythonProject\Robot_framework\Login.robot
 - Start / End / Elapsed: 20250819 15:39:12.945 / 20250819 15:40:53.564 / 00:01:40.619
 - Status: 1 test total, 1 passed, 0 failed, 0 skipped
- TEST** ERP Login With Popup Handling (00:01:40.163)

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