Steps to Automated Testing

1. Understand the requirement

2. Set up the Test Environment

3. Identify Test Scenarios

4. Write Automated Test Scripts

5. Run Tests and Analyze Result

6. Report Bugs

7. Maintain and reuse scripts

**✅ Step-by-Step Guide for Automated Testing**

**1. Understand the Requirements**

* Go through the **specification document** or user stories related to your module.
* Understand **what the module should do**, and identify the **functionalities and expected behavior**.

**2. Set Up the Test Environment**

* Make sure all the tools needed are installed:
  + **IDE** (e.g., PyCharm, VSCode, Eclipse)
  + **Test framework** (e.g., Selenium, Robot Framework, JUnit, PyTest)
  + **Browser Drivers** (for web testing, like ChromeDriver)
* Configure necessary libraries and **dependencies**.

**3. Identify Test Scenarios**

* Break the module into **test cases**:
  + Positive test cases (valid inputs)
  + Negative test cases (invalid inputs)
  + Edge cases (boundary values)
* Example: If testing a login module, write test cases for:
  + Valid login
  + Invalid username
  + Invalid password
  + Empty fields

**4. Write Automated Test Scripts**

* Use the test automation tool provided (e.g., Selenium, Robot Framework).
* Follow this structure:
  + **Setup** (open browser, navigate to module)
  + **Test Steps** (enter data, click buttons)
  + **Assertions** (check expected result)
  + **Teardown** (close browser or clean up)

If you're using Robot Framework, test case will look like:

robot

Copy code

\*\*\* Test Cases \*\*\*

Valid Login

Open Browser https://example.com Chrome

Input Text username\_field testuser

Input Text password\_field testpass

Click Button login\_button

Page Should Contain Welcome, testuser

Close Browser

**5. Run Tests and Analyze Results**

* Run your scripts using the test runner.
* Check the **test reports/logs** to see if all test cases passed.
* If a test fails, check:
  + Was the script incorrect?
  + Is there a bug in the application?

**6. Report Bugs (if found)**

* Use your company’s bug tracking tool (e.g., Jira, Bugzilla).
* Include:
  + Steps to reproduce
  + Screenshots or logs
  + Expected vs Actual result

**7. Maintain and Reuse Scripts**

* Make scripts **modular** for reuse (e.g., login function).
* Update test scripts if the application changes (regression testing).

**💡 Bonus Tips for Freshers**

* Ask **clarifying questions** to seniors or developers if you're unsure.
* Start with **manual testing** before automation to understand flow.
* Write **clean, readable, and well-commented code**.

Testing

1. Browser Properly open or not