# Test manual and Test automated

## 1-Test manual

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**

A **test manual** (or more accurately, **manual testing**) is a type of software testing done **by a human tester** without using automation tools to execute the test cases.

### 🖐️ Definition

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**

Here’s what it means in simple terms:

* It’s the process of **checking the software manually**, step by step, following written or understood test cases.
* A tester acts as the end-user and tries out different inputs, clicks, navigation, and checks whether the software behaves as expected.
* No scripts or automation tools run the tests — it’s all human interaction.

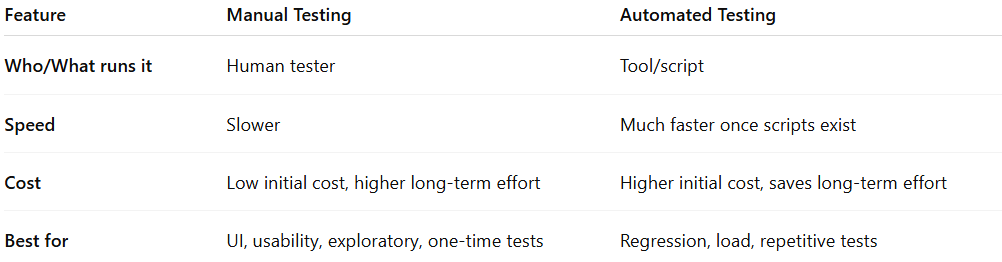
### 🔹 Key Characteristics

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**

* **Human-driven**: A tester executes the tests.
* **No coding/scripts required**: Done without automated frameworks or tools.
* **Useful for**:
  + Exploratory testing.
  + Usability testing (user-friendliness).
  + Small projects or where automation is too costly or unnecessary.

### 🆚 Manual Testing vs. Automated Testing

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**



| **Feature** | **Manual Testing** | **Automated Testing** |
| --- | --- | --- |
| **Who/What runs it** | Human tester | Tool/script |
| **Speed** | Slower | Much faster once scripts exist |
| **Cost** | Low initial cost, higher long-term effort | Higher initial cost, saves long-term effort |
| **Best for** | UI, usability, exploratory, one-time tests | Regression, load, repetitive tests |

### 💡 Simple example:

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**  
You have a login page. In manual testing, the tester:

1. Opens the browser.
2. Types username and password.
3. Clicks “Login”.
4. Checks whether the expected page loads correctly.
5. Repeats for different input scenarios (valid, invalid, empty fields).

### Test Manual Document

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**

Perfect 👍 Now we’re talking about something different: a **Test Manual Document** (sometimes just called “Test Manual”). This is not about doing tests manually, but about **a document** that explains **how** to do the testing, usually in a clear, structured way. Here’s what it means:

#### 📄 Definition

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**

* A **formal document** that describes **testing procedures**, **steps**, **environments**, and **data** needed to perform tests on a software system.
* It’s like an instruction book for testers — anyone can read it and know **what to test, how to test, and what results to expect**.

#### 🔹 What it usually contains

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**

1. **Introduction / Purpose**
   * Why this test manual exists and what part of the software it covers.
2. **Scope**
   * Which features, modules, or versions are included/excluded.
3. **Test Environment**
   * Hardware, software, network, or special setups required.
4. **Test Cases and Procedures**
   * Step-by-step instructions:
     + Preconditions (what should be ready before starting).
     + Test steps (what to click, enter, or do).
     + Expected results (what should happen).
5. **Test Data**
   * Specific inputs to use (e.g., usernames, passwords, file types).
6. **Roles and Responsibilities**
   * Who performs each test or approves results.
7. **Reporting**
   * How to log results, where to report defects.
8. **References**
   * Related requirement documents, design specs, or user stories.

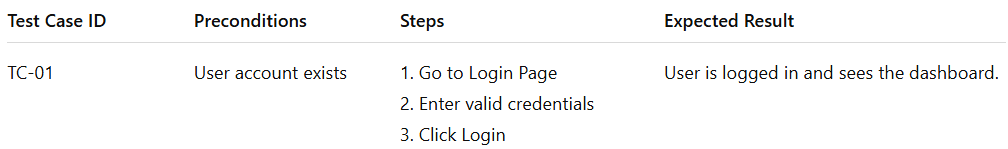
#### 🔹 Purpose

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**

* Ensures **consistency** — all testers follow the same steps.
* Helps **new testers** quickly understand how to test the system.
* Provides **evidence** for audits or certifications.

#### 💡 Example (Very simple):

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**



| **Test Case ID** | **Preconditions** | **Steps** | **Expected Result** |
| --- | --- | --- | --- |
| TC-01 | User account exists | 1. Go to Login Page2. Enter valid credentials3. Click Login | User is logged in and sees the dashboard. |

### Summary

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**

So:

* **Manual testing** = doing tests by hand.
* **Test manual** (document) = a guide or “instruction book” for how to perform those tests.

## 2-Test automated

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**

Great follow-up 👍 Now we’re talking about **Automated Testing** (sometimes people just say “test automated” or “automated tests”).

### 🤖 Definition

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**

Here’s the meaning in simple terms:

* A **software testing approach** where tests are executed **by tools or scripts** instead of by a human manually clicking and typing.
* You write test scripts (programs) once, then a tool (like Selenium, JUnit, Cypress, Playwright, etc.) runs those tests automatically whenever you want.

### 🔹 Key Points

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**

* **Runs without human intervention** after setup.
* **Fast and repeatable** — great for checking the same thing many times (e.g., after every code change).
* Can run **day or night**, or as part of CI/CD pipelines.
* Used for **regression testing** (checking that new changes didn’t break old features).

### 🔹 Advantages

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**

1. **Speed** — much faster than manual testing.
2. **Accuracy** — no human error in following steps.
3. **Coverage** — can test many combinations and scenarios quickly.
4. **Cost effective** over time — initial setup is work, but then it saves labor.

### 🔹 Limitations

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**

* **Initial effort** is high — you have to write the scripts and maintain them.
* **Not good** for everything — some tests (like exploratory testing or checking “look & feel”) still need a human.
* **Tools** need updates when the system changes (maintenance overhead).

### 💡 Example:

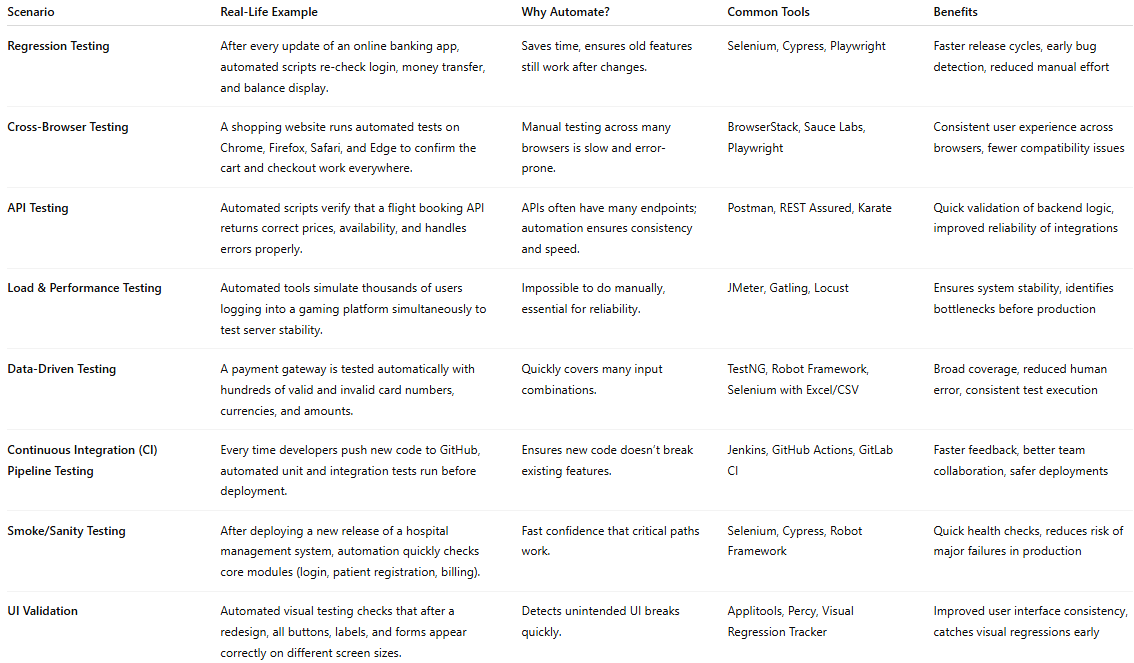
**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**  
Instead of a tester opening a browser and logging in manually, an automated script might:

1. Open the browser.
2. Go to the login page.
3. Enter username/password from a data file.
4. Click login.
5. Verify the dashboard loads.

All in a few seconds — and it can repeat this for 50 test accounts automatically.

### real-life examples of test automation scenarios

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**



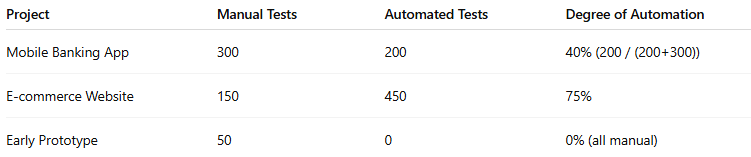
| **Scenario** | **Real-Life Example** | **Why Automate?** | **Common Tools** | **Benefits** |
| --- | --- | --- | --- | --- |
| **Regression Testing** | After every update of an online banking app, automated scripts re-check login, money transfer, and balance display. | Saves time, ensures old features still work after changes. | Selenium, Cypress, Playwright | Faster release cycles, early bug detection, reduced manual effort |
| **Cross-Browser Testing** | A shopping website runs automated tests on Chrome, Firefox, Safari, and Edge to confirm the cart and checkout work everywhere. | Manual testing across many browsers is slow and error-prone. | BrowserStack, Sauce Labs, Playwright | Consistent user experience across browsers, fewer compatibility issues |
| **API Testing** | Automated scripts verify that a flight booking API returns correct prices, availability, and handles errors properly. | APIs often have many endpoints; automation ensures consistency and speed. | Postman, REST Assured, Karate | Quick validation of backend logic, improved reliability of integrations |
| **Load & Performance Testing** | Automated tools simulate thousands of users logging into a gaming platform simultaneously to test server stability. | Impossible to do manually, essential for reliability. | JMeter, Gatling, Locust | Ensures system stability, identifies bottlenecks before production |
| **Data-Driven Testing** | A payment gateway is tested automatically with hundreds of valid and invalid card numbers, currencies, and amounts. | Quickly covers many input combinations. | TestNG, Robot Framework, Selenium with Excel/CSV | Broad coverage, reduced human error, consistent test execution |
| **Continuous Integration (CI) Pipeline Testing** | Every time developers push new code to GitHub, automated unit and integration tests run before deployment. | Ensures new code doesn’t break existing features. | Jenkins, GitHub Actions, GitLab CI | Faster feedback, better team collaboration, safer deployments |
| **Smoke/Sanity Testing** | After deploying a new release of a hospital management system, automation quickly checks core modules (login, patient registration, billing). | Fast confidence that critical paths work. | Selenium, Cypress, Robot Framework | Quick health checks, reduces risk of major failures in production |
| **UI Validation** | Automated visual testing checks that after a redesign, all buttons, labels, and forms appear correctly on different screen sizes. | Detects unintended UI breaks quickly. | Applitools, Percy, Visual Regression Tracker | Improved user interface consistency, catches visual regressions early |

### degree of test automation

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**

**Degree of Test Automation**

* **Definition**:  
  The **extent (percentage or level)** to which testing activities are automated compared to being done manually.  
  In other words, it shows **how much** of your testing is performed by automated tools instead of human testers.
* **Why It Matters**:  
  It helps organizations measure:
  + Efficiency (less manual effort)
  + Coverage (more scenarios tested automatically)
  + ROI (return on investment for automation tools)
  + Planning (how much still needs human/manual testing)
* **Examples**:



| **Project** | **Manual Tests** | **Automated Tests** | **Degree of Automation** |
| --- | --- | --- | --- |
| Mobile Banking App | 300 | 200 | 40% (200 / (200+300)) |
| E-commerce Website | 150 | 450 | 75% |
| Early Prototype | 50 | 0 | 0% (all manual) |

* **Common Levels** (not official but practical):
  + **Low** → <30% automated (mostly manual testing)
  + **Medium** → 30–70% automated (mix of manual & automation)
  + **High** → >70% automated (majority of tests run by scripts/tools)
* **Note**:  
  It’s rarely 100% automated — because **exploratory testing**, **usability**, and **ad-hoc checks** still need humans.

### Summary

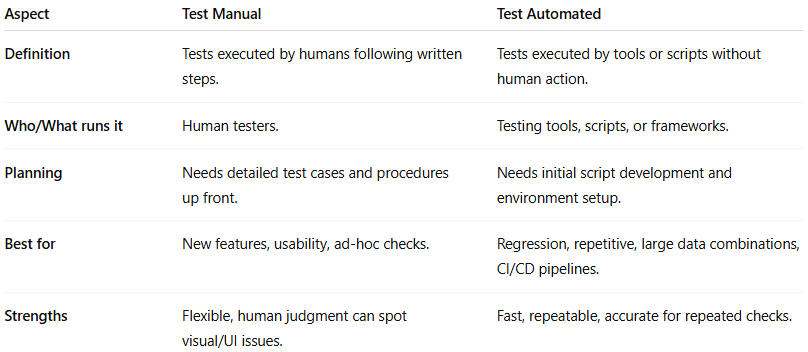
**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**

So, in short:

| **Aspect** | **Manual Testing** | **Automated Testing** |
| --- | --- | --- |
| Performed by | Human | Tool / Script |
| Speed | Slower | Much faster |
| Cost (long term) | Higher | Lower (after setup) |
| Use cases | Exploratory, usability | Regression, repetitive checks |

## Different approaches of testing

**# Source**: **Chatgot (GPT-5-Standard) at [9/4/2025]**



| **Aspect** | **Test Manual** | **Test Automated** |
| --- | --- | --- |
| **Definition** | Tests executed by humans following written steps. | Tests executed by tools or scripts without human action. |
| **Who/What runs it** | Human testers. | Testing tools, scripts, or frameworks. |
| **Planning** | Needs detailed test cases and procedures up front. | Needs initial script development and environment setup. |
| **Best for** | New features, usability, ad-hoc checks. | Regression, repetitive, large data combinations, CI/CD pipelines. |
| **Strengths** | Flexible, human judgment can spot visual/UI issues. | Fast, repeatable, accurate for repeated checks. |