

Software Testing Test Case Guide PDF Page

Software Testing Test Case Guide PDF Page: Content + <head> (Uncodemy)

FINAL URL: <https://uncodemy.com/software-testing-test-case-guide-pdf>

Publish this page at exactly this URL (non-www, no trailing slash). Canonical, og:url and sitemap entry must match it.

Prepared: 26 October 2026 · **Primary keyword:** software testing test case guide pdf · **Search volume:** not checked (keyword tool out of credits); verify in Search Console.

Read first (5 flags):

- Your course PDF title, page count and size were not shared.** Download cards are fill-in slots ([CONFIRM]). Card 1 is your course PDF; cards 2 and 3 only if you have more files.
- One unique file per button.** The same course PDF repeated on many pages is fine as a resource, but each page's text must stay unique and useful without the file.
- Free or gated?** Not stated. The page says "Download", not "Free". If a form is used, add a consent checkbox and Privacy Policy link (DPDP Act 2023).
- No made-up stats or testimonials;** no placement or salary claims without a named, dated source.
- Author and reviewer are placeholders;** use real named trainers and one real "Last reviewed" date.

PART 1: Page plan

Item	Decision
Final URL	https://uncodemy.com/software-testing-test-case-guide-pdf
Intent (anti-cannibalisation)	WRITING TEST CASES (template, techniques, example). Test scenarios are on their own page; bug reports are on the bug reporting page.
Why not thin	Explainer-style sections, FAQs and cross-links make the page useful even without a download
Schema	WebSite, Organization, Breadcrumb, CollectionPage, FAQPage (only FAQs visible on the page). PDF <code>ItemList</code> provided commented, to enable after titles are final
Share image	https://uncodemy.com/image/logo-og.jpg for now; custom 1200x630 card recommended

PART 2: Page content (copy-paste ready)

SEO Title (47 chars):

Software Testing Test Case Guide PDF | Uncodemy

Meta Description (131 chars):

Download the software testing test case guide PDF from Uncodemy: a standard template, design techniques and a worked login example.

H1: Software Testing Test Case Guide PDF Downloads **Breadcrumb:** Home > Software Testing Test Case Guide PDF **Final URL:** <https://uncodemy.com/software-testing-test-case-guide-pdf>

Byline: Prepared by the Uncodemy trainer team [CONFIRM: author name and role] · Reviewed by [CONFIRM: reviewer name and role] · Last reviewed: [DATE]

Quick Answer

This page offers Uncodemy's test case guide as a PDF download. A test case is a documented set of inputs, steps and expected results that checks one specific behaviour of the software. The guide gives a standard template, explains design techniques such as boundary value analysis and equivalence partitioning, and works through a login example you can adapt.

Download Software Testing Test Case Guide PDF

[CONFIRM: free or gated, and whether details are needed before download]

PDF 1 (main): [CONFIRM TITLE: your Software Testing course PDF]
[CONFIRM: one line on what it covers and who it is for] · [N] pages · [X] MB · PDF · Updated [DATE]
[Download PDF] → <https://uncodemy.com/pdf/software-testing-test-case-guide.pdf> (your course PDF, uploaded under this file name)

PDF 2 (optional): [CONFIRM TITLE]
Only if you have a second PDF that genuinely fits this topic. **[Download PDF]** → <https://uncodemy.com/pdf/software-testing-test-case-guide-2.pdf>

PDF 3 (optional): [CONFIRM TITLE]

Same rule. [Download PDF] → <https://uncodemy.com/pdf/software-testing-test-case-guide-3.pdf>

Each button must open its own file. Do not point several buttons at one file. Delete unused optional cards.

What is a test case?

A test case describes one check of the software in enough detail that anyone on the team can run it and reach the same verdict. It states the starting conditions, the exact steps, the data to use and the result that should happen. Test cases serve three purposes. They guide testers so that nothing important is forgotten. They make testing repeatable, so the same check can be run on every release. And they are evidence of what was covered, which matters when someone asks whether a feature was tested. A good test case is small, specific and independent: it checks one thing, it does not depend on another case having run first, and its expected result is clear enough that a pass or fail is not a matter of opinion.

A standard test case template

Field	What to write	Example
Test case ID	A unique identifier	TC-LOGIN-001
Title	One line saying what is checked	Login succeeds with valid credentials
Module and requirement	Where it belongs and what it traces to	Login; REQ-12
Preconditions	State needed before starting	User account exists and is active
Test data	Exact values to use	Email: test.user@example.com ; valid password
Steps	Numbered actions	1. Open login page 2. Enter email 3. Enter password 4. Click Login
Expected result	What should happen	User lands on the dashboard and sees their name
Actual result and status	Filled in at execution	Pass or Fail, with notes
Priority and type	Importance and category	High; Functional

Writing good steps and expected results

Write steps as short, imperative sentences that describe actions and name the screen elements involved. Avoid vague wording such as check that it works or verify the page. The expected result must be observable and specific: not login should work but the dashboard page opens, the welcome message shows the user's name and no error appears. Put one verification point at the end of each case where possible; a case that tries to check ten things makes failures hard to diagnose. Keep test data separate and explicit, never assume the tester will invent a suitable value.

Test design techniques

Writing a test case for every possible input is impossible, so testers use techniques that choose a small set of inputs most likely to reveal defects. Equivalence partitioning divides inputs into groups that the software should treat the same way and tests one value from each group. Boundary value analysis focuses on the edges of those groups, where mistakes are most common. Decision tables list combinations of conditions and the resulting actions, which is useful for rules with several factors. State transition testing checks that a system moves correctly between states, such as an order moving from placed to paid to shipped. Error guessing uses experience to try inputs that have caused trouble before.

Boundary value example: password length 8 to 20 characters

Input length	Class	Expected result
7	Just below the lower boundary	Rejected with a length error
8	On the lower boundary	Accepted
9	Just above the lower boundary	Accepted
19	Just below the upper boundary	Accepted
20	On the upper boundary	Accepted
21	Just above the upper boundary	Rejected with a length error

Sample test cases for a login page

ID	Title	Test data	Expected result
TC-LOGIN-001	Valid login	Registered email and correct password	Dashboard opens
TC-LOGIN-002	Wrong password	Registered email and wrong password	Error: invalid credentials; user stays on login page
TC-LOGIN-003	Unregistered email	Unknown email and any password	Generic error; no hint about which field is wrong
TC-LOGIN-004	Empty fields	Both fields blank	Validation messages for both fields
TC-LOGIN-005	Invalid email format	Text without an at sign	Format validation message
TC-LOGIN-006	Password is masked	Type any password	Characters are hidden
TC-LOGIN-007	Account lockout	Several wrong attempts in a row	Account locks or a delay applies, according to the requirement
TC-LOGIN-008	Leading and trailing spaces in email	Valid email with spaces	Behaviour matches the requirement; no crash

Positive and negative test cases

- Positive cases confirm that valid input produces the correct result.

- Negative cases confirm that invalid input is handled safely with a clear message.
- Boundary cases check the edges of allowed ranges.
- A balanced suite includes all three; a suite of only positive cases misses most real defects.

Qualities of a good test case

- **Clear:** anyone can follow it without asking questions.
- **Atomic:** it checks one behaviour.
- **Independent:** it does not rely on another case's result.
- **Traceable:** it links to a requirement or user story.
- **Repeatable:** it gives the same result when run again on the same build.
- **Maintainable:** it is easy to update when the requirement changes.

Common mistakes when writing test cases

- **Vague expected results** such as it should work.
- **Combining many checks** in one case.
- **Missing preconditions and test data.**
- **Copying cases without adapting** them to the feature.
- **Never reviewing or updating** cases as the product changes.

Review and maintenance

A test case suite is a living asset. Have a colleague review new cases against the requirement, remove duplicates and retire cases for features that no longer exist. After each release add cases for defects that escaped, so that the same problem is caught next time. Tag cases by priority so that a short run of the most important ones can be used when time is tight.

Related Uncodemy resources

- [Software Testing Test Scenario Guide PDF](#)
- [Software Testing Bug Reporting Guide PDF](#)
- [Software Testing Functional Testing PDF](#)
- [Software Testing Test Plan Guide PDF](#)

Learn with a trainer [use only the line that is true]

- **Software Testing course:** [CONFIRM URL]. [CONFIRM URL] Link only if this course page is live; otherwise use <https://uncodemy.com/all-courses>

(One soft call to action: course link or free demo. No pop-up.)

About these PDFs

These PDFs are study material for learning and revision. They are not a certificate, a degree or a promise of a job, and technology changes, so check official documentation for the latest details. Spot an error? Write to [CONFIRM: feedback_email] and the trainer team will review it.

Frequently Asked Questions

What is a test case in software testing?

A test case is a documented set of conditions, steps, data and expected results used to verify one specific behaviour of the software.

What should a test case contain?

A unique ID, title, preconditions, test data, steps, expected result, actual result, status and priority, with a link to the requirement.

What is the difference between a positive and a negative test case?

A positive case uses valid input and expects success. A negative case uses invalid input and expects a safe, clear error.

What is boundary value analysis?

A technique that tests values at and just beyond the edges of allowed ranges, where defects are most likely.

How many test cases are enough?

There is no fixed number. Use design techniques to cover the important behaviours and risks with the fewest meaningful cases.

Are these PDFs free to download?

[CONFIRM: state clearly whether the PDFs are free, or whether a short form is needed first.]

The last FAQ contains [CONFIRM], so it is NOT in the FAQ schema. Fill it and keep it on the page, or delete it.

PART 3: <head> section (copy-paste ready)

Final URL: <https://uncodemy.com/software-testing-test-case-guide-pdf>

```
<!-- =====
UNCODEMY <head> | FINAL URL: https://uncodemy.com/software-testing-test-case-guide-pdf
Prepared 26 October 2026. Paste into the page <head>; keep your site-wide favicon, fonts, analytics.
BEFORE GO-LIVE: (1) set dateModified in the JSON-LD to the real publish date;
(2) replace og:image with a 1200x630 share card when available;
(3) once PDF titles are final, uncomment the ItemList block at the bottom.
===== -->

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```

<title>Software Testing Test Case Guide PDF | Uncodemy</title>
<meta name="description" content="Download the software testing test case guide PDF from Uncodemy: a standard template, design techniques and a worked login
example.">
<link rel="canonical" href="https://uncodemy.com/software-testing-test-case-guide-pdf">
<meta name="robots" content="index, follow, max-snippet:-1, max-image-preview:large, max-video-preview:-1">
<meta name="author" content="Uncodemy">
<meta name="publisher" content="Uncodemy Edutech Pvt. Ltd.">
<meta name="language" content="en-IN">
<link rel="alternate" hreflang="en-in" href="https://uncodemy.com/software-testing-test-case-guide-pdf">
<link rel="alternate" hreflang="x-default" href="https://uncodemy.com/software-testing-test-case-guide-pdf">

<!-- Open Graph -->
<meta property="og:type" content="website">
<meta property="og:site_name" content="Uncodemy">
<meta property="og:locale" content="en-IN">
<meta property="og:url" content="https://uncodemy.com/software-testing-test-case-guide-pdf">
<meta property="og:title" content="Software Testing Test Case Guide PDF | Uncodemy">
<meta property="og:description" content="Download the software testing test case guide PDF from Uncodemy: a standard template, design techniques and a worked
login example.">
<meta property="og:image" content="https://uncodemy.com/image/logo-og.jpg">
<meta property="og:image:alt" content="Uncodemy Software Testing Test Case Guide PDF study material">

<!-- X (Twitter) -->
<meta name="twitter:card" content="summary_large_image">
<meta name="twitter:site" content="@uncodemy">
<meta name="twitter:title" content="Software Testing Test Case Guide PDF | Uncodemy">
<meta name="twitter:description" content="Download the software testing test case guide PDF from Uncodemy: a standard template, design techniques and a worked
login example.">
<meta name="twitter:image" content="https://uncodemy.com/image/logo-og.jpg">
<meta name="twitter:image:alt" content="Uncodemy Software Testing Test Case Guide PDF study material">

<!-- Structured data: one @graph (WebSite, Organization, Breadcrumb, CollectionPage, FAQPage) -->
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@graph": [
    {
      "@type": "WebSite",
      "@id": "https://uncodemy.com/#website",
      "url": "https://uncodemy.com/",
      "name": "Uncodemy",
      "publisher": {
        "@id": "https://uncodemy.com/#organization"
      },
      "inLanguage": "en-IN"
    },
    {
      "@type": "EducationalOrganization",
      "@id": "https://uncodemy.com/#organization",
      "name": "Uncodemy",
      "legalName": "Uncodemy Edutech Pvt. Ltd.",
      "url": "https://uncodemy.com/",
      "logo": {
        "@type": "ImageObject",
        "url": "https://uncodemy.com/uc-layout/img/Logo.webp"
      },
      "email": "info@uncodemy.com",
      "identifier": {
        "@type": "PropertyValue",
        "name": "CIN",
        "value": "U85499UP2024PTC195902"
      },
      "contactPoint": [
        {
          "@type": "ContactPoint",
          "telephone": "+91-9818366550",
          "contactType": "admissions",
          "areaServed": "IN",
          "availableLanguage": [
            "en",
            "hi"
          ]
        }
      ],
      "sameAs": [
        "https://x.com/uncodemy",
        "https://www.facebook.com/uncodemyofficial/",
        "https://www.linkedin.com/company/uncodemy/",
        "https://www.instagram.com/uncodemyofficial/"
      ]
    },
    {
      "@type": "BreadcrumbList",
      "@id": "https://uncodemy.com/software-testing-test-case-guide-pdf#breadcrumb",
      "itemListElement": [
        {
          "@type": "ListItem",
          "position": 1,
          "name": "Home",
          "item": "https://uncodemy.com/"
        },
        {
          "@type": "ListItem",
          "position": 2,
          "name": "Software Testing Test Case Guide PDF",
          "item": "https://uncodemy.com/software-testing-test-case-guide-pdf"
        }
      ]
    },
    {
      "@type": "CollectionPage",
      "@id": "https://uncodemy.com/software-testing-test-case-guide-pdf#webpage",
      "url": "https://uncodemy.com/software-testing-test-case-guide-pdf",
      "name": "Software Testing Test Case Guide PDF Downloads",
      "description": "Download the software testing test case guide PDF from Uncodemy: a standard template, design techniques and a worked login example.",
      "inLanguage": "en-IN",
      "isPartOf": {
        "@id": "https://uncodemy.com/#website"
      },
      "publisher": {
        "@id": "https://uncodemy.com/#organization"
      }
    }
  ]
}

```

```

    },
    "breadcrumb": {
      "@id": "https://uncodemy.com/software-testing-test-case-guide-pdf#breadcrumb"
    },
    "about": {
      "@type": "Thing",
      "name": "Software Testing Test Case Guide"
    },
    "primaryImageOfPage": {
      "@type": "ImageObject",
      "url": "https://uncodemy.com/image/logo-og.jpg"
    },
    "dateModified": "2026-10-26"
  },
  {
    "@type": "FAQPage",
    "@id": "https://uncodemy.com/software-testing-test-case-guide-pdf#faq",
    "inLanguage": "en-IN",
    "isPartOf": {
      "@id": "https://uncodemy.com/software-testing-test-case-guide-pdf#webpage"
    },
    "mainEntity": [
      {
        "@type": "Question",
        "name": "What is a test case in software testing?",
        "acceptedAnswer": {
          "@type": "Answer",
          "text": "A test case is a documented set of conditions, steps, data and expected results used to verify one specific behaviour of the software."
        }
      },
      {
        "@type": "Question",
        "name": "What should a test case contain?",
        "acceptedAnswer": {
          "@type": "Answer",
          "text": "A unique ID, title, preconditions, test data, steps, expected result, actual result, status and priority, with a link to the requirement."
        }
      },
      {
        "@type": "Question",
        "name": "What is the difference between a positive and a negative test case?",
        "acceptedAnswer": {
          "@type": "Answer",
          "text": "A positive case uses valid input and expects success. A negative case uses invalid input and expects a safe, clear error."
        }
      },
      {
        "@type": "Question",
        "name": "What is boundary value analysis?",
        "acceptedAnswer": {
          "@type": "Answer",
          "text": "A technique that tests values at and just beyond the edges of allowed ranges, where defects are most likely."
        }
      },
      {
        "@type": "Question",
        "name": "How many test cases are enough?",
        "acceptedAnswer": {
          "@type": "Answer",
          "text": "There is no fixed number. Use design techniques to cover the important behaviours and risks with the fewest meaningful cases."
        }
      }
    ]
  }
]
}
</script>

<!-- ADD AFTER THE PDF TITLES ARE FINAL (remove this wrapper, fill every [...] value):
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "ItemList",
  "@id": "https://uncodemy.com/software-testing-test-case-guide-pdf#pdfs",
  "name": "Software Testing Test Case Guide PDF downloads",
  "numberOfItems": 3,
  "itemListElement": [
    {
      "@type": "ListItem",
      "position": 1,
      "item": {
        "@type": "DigitalDocument",
        "name": "[PDF 1 TITLE]",
        "description": "[PDF 1 ONE-LINE DESCRIPTION]",
        "encodingFormat": "application/pdf",
        "url": "https://uncodemy.com/pdf/software-testing-test-case-guide.pdf",
        "inLanguage": "en",
        "publisher": {
          "@id": "https://uncodemy.com/#organization"
        }
      }
    },
    {
      "@type": "ListItem",
      "position": 2,
      "item": {
        "@type": "DigitalDocument",
        "name": "[PDF 2 TITLE]",
        "description": "[PDF 2 ONE-LINE DESCRIPTION]",
        "encodingFormat": "application/pdf",
        "url": "https://uncodemy.com/pdf/software-testing-test-case-guide-2.pdf",
        "inLanguage": "en",
        "publisher": {
          "@id": "https://uncodemy.com/#organization"
        }
      }
    },
    {
      "@type": "ListItem",
      "position": 3,
      "item": {
        "@type": "DigitalDocument",

```

```

    "name": "[PDF 3 TITLE]",
    "description": "[PDF 3 ONE-LINE DESCRIPTION]",
    "encodingFormat": "application/pdf",
    "url": "https://uncodemy.com/pdf/software-testing-test-case-guide-3.pdf",
    "inLanguage": "en",
    "publisher": {
      "@id": "https://uncodemy.com/#organization"
    }
  }
}
}
</script>
-->

```

DEVELOPER CHECKLIST

- ☐ Final URL live at <https://uncodemy.com/software-testing-test-case-guide-pdf>; canonical, og:url and sitemap identical.
- ☐ Each button: `<a href="/pdf/software-testing-test-case-guide.pdf" download type="application/pdf">`, its own file, size shown.
- ☐ If a form precedes download: consent checkbox, Privacy Policy link, nothing pre-ticked.
- ☐ One `<h1>`; real HTML tables; code samples in `<pre><code>`.
- ☐ Remove any `<meta name="keywords">` from the template.
- ☐ Test in Google Rich Results Test and validator.schema.org after publishing.
- ☐ Add to sitemap and any PDF/resources hub.

STILL NEEDED FROM YOU

#	Item
1	Course PDF title, description, page count, size, and the file URL for each page
2	Free or gated (and form fields if gated)
3	Author, reviewer, publish and last-reviewed dates
4	Course URL for the CTA (lines marked <code>[CONFIRM]</code>)
5	Feedback email
6	Confirmation that Uncodemy owns the rights to every PDF