

# Test Case Template

# Test Case Template

Project: {{PROJECT\_NAME}} Version: {{VERSION}} Date: {{DATE}}  
Author: {{AUTHOR}} Status: Draft | In Review | Approved Reviewers:  
{{REVIEWERS}}

## Document History

Version	Date	Author	Changes
0.1	{{DATE}}	{{AUTHOR}}	Initial draft

## 1. Test Case ID Format & Naming Convention

Format: TC-{{MODULE\_CODE}}-{{SEQUENCE}}

Part	Description	Example
TC	Test Case prefix (always TC)	TC
MODULE_CODE	2-4 letter module abbreviation	AUTH, CART, PAY, USR
SEQUENCE	3-digit zero-padded number	001, 042, 100

### Examples:

- TC-AUTH-001 — Authentication module, first test case
- TC-PAY-015 — Payment module, 15th test case
- TC-CART-003 — Shopping cart module, 3rd test case

Related requirement ID: Link to REQ-{{ID}} or user story {{TICKET\_ID}}

## 2. Test Suite Organization

Suite ID	Suite Name	Module	Test Cases	Owner
TS-AUTH	Authentication	Auth module	TC-AUTH-001 to TC-AUTH-XXX	{{OWNER}}
TS-CART	Shopping Cart	Cart module	TC-CART-001 to TC-CART-XXX	{{OWNER}}
TS-PAY	Payment Processing	Payment module	TC-PAY-001 to TC-PAY-XXX	{{OWNER}}
TS-SMOKE	Smoke Tests	All critical paths	TC-SMK-001 to TC-SMK-XXX	{{OWNER}}
TS-REG	Regression Suite	Full application	All high-priority TCs	{{OWNER}}

## 3. Individual Test Case Format

Test Case: TC-{{MODULE}}-{{SEQ}}

Field	Value
<b>ID</b>	TC-{{MODULE}}-{{SEQ}}
<b>Title</b>	{{CONCISE_DESCRIPTION}}
<b>Description</b>	{{FULL_DESCRIPTION}}
<b>Module / Feature</b>	{{MODULE_NAME}}
<b>Priority</b>	{{PRIORITY}}
<b>Type</b>	{{TYPE}}
<b>Requirement</b>	{{REQ_ID}} / {{STORY_ID}}
<b>Automation Status</b>	{{STATUS}}
<b>Automation ID</b>	{{AUTOMATION_ID}}

### Preconditions

1. {{PRECONDITION\_1}}
2. {{PRECONDITION\_2}}
3. {{PRECONDITION\_3}}

## Test Data

Field	Value	Notes
Email	{{TEST_EMAIL}}	Test account, pre-created
Password	{{TEST_PASSWORD}}	Stored in test vault
{{DATA_FIELD}}	{{VALUE}}	{{NOTE}}

## Test Steps

Step	Action	Expected Result
1	{{ACTION_1}}	{{EXPECTED_1}}
2	{{ACTION_2}}	{{EXPECTED_2}}
3	{{ACTION_3}}	{{EXPECTED_3}}
4	{{ACTION_4}}	{{EXPECTED_4}}

## Expected Final Result

{{OVERALL\_EXPECTED\_RESULT}}

## Post-conditions

- {{POSTCONDITION\_1}}
- {{POSTCONDITION\_2}}

## Notes / Edge Cases

- {{NOTE\_1}}
- Related: TC-{{RELATED\_ID}}

# 4. Priority Definitions

Priority	Definition	Examples
<b>Critical</b>	Core functionality; system unusable without it	Login, payment, data persistence
<b>High</b>	Important feature; significant user impact if broken	Search, notifications, user profile
<b>Medium</b>	Standard feature; workaround exists	Export, advanced filters, preferences
<b>Low</b>	Minor feature, cosmetic, or edge case	Tooltips, sorting preferences, animations

---

## 5. Test Case Type Definitions

Type	Description
<b>Functional</b>	Verifies a feature works as specified
<b>Regression</b>	Verifies previously working functionality still works
<b>Smoke</b>	Fast check of most critical paths (subset for quick confidence)
<b>Boundary</b>	Tests at the edges of valid input (min, max, empty)
<b>Negative</b>	Tests invalid input, error conditions, unauthorized access
<b>Performance</b>	Verifies response time, throughput under defined load
<b>Security</b>	Verifies access controls, injection resistance, auth
<b>Accessibility</b>	Verifies WCAG compliance, keyboard navigation, screen readers

---

## 6. Batch Test Execution Template

**Test Run:** {{RUN\_ID}} **Environment:** {{ENVIRONMENT}} **Build / Version:** {{VERSION}}  
**Tester:** {{TESTER}} **Date:** {{DATE}}

Test Case ID	Title	Priority	Result	Actual Result / Notes	Defect ID
TC-{{MODULE}}-001	{{TITLE}}	Critical	Pass / Fail / Blocked / Skip	{{NOTES}}	{{DEFECT_ID}}
TC-{{MODULE}}-002	{{TITLE}}	High			

### Summary:

- Total: {{TOTAL}}
  - Passed: {{PASSED}}
  - Failed: {{FAILED}}
  - Blocked: {{BLOCKED}}
  - Skipped: {{SKIPPED}}
  - Pass rate: {{PASS\_RATE}}%
-

# 7. Test Execution Log Format

Timestamp	Test Case ID	Tester	Environment	Build	Result	Duration	Notes
{{TIMESTAMP}}	TC-{{MODULE}}-{{SEQ}}	{{TESTER}}	staging	{{BUILD}}	Pass	{{DURATION}}s	

# 8. Defect Linking

**Defect format:** `BUG-{{ID}}` (in `{{BUG_TRACKER}}`)

Test Case	Defect ID	Severity	Status	Fixed In
TC-{{MODULE}}-{{SEQ}}	BUG-{{ID}}	{{SEVERITY}}	{{STATUS}}	{{VERSION}}

**Defect fields required:**

- Steps to reproduce (reference test case ID)
- Expected vs actual behavior
- Environment + build version
- Screenshot / screen recording
- Severity and priority

# Example Test Cases

## Test Case: TC-AUTH-001

Field	Value
<b>ID</b>	TC-AUTH-001
<b>Title</b>	User can log in with valid email and password
<b>Description</b>	Verifies that a registered user can successfully authenticate using correct credentials and is redirected to the dashboard.
<b>Module / Feature</b>	Authentication — Login
<b>Priority</b>	Critical

Field	Value
Type	Functional
Requirement	REQ-AUTH-001
Automation Status	Automated
Automation ID	tests/e2e/auth/login.spec.ts:valid-credentials

## Preconditions

- User account with email `qa-test@example.com` exists and is verified
- User is not logged in (no active session)
- Application is accessible at `{{APP_URL}}/login`

## Test Data

Field	Value	Notes
Email	<code>qa-test@example.com</code>	Pre-created test account
Password	Retrieved from test vault	Never hardcoded

## Test Steps

Step	Action	Expected Result
1	Navigate to <code>{{APP_URL}}/login</code>	Login page loads with email field, password field, and login button
2	Enter <code>qa-test@example.com</code> in email field	Email value visible in field
3	Enter valid password in password field	Password masked, not visible
4	Click "Log In" button	Loading indicator shown, network request initiated
5	Wait for response	Redirect to <code>/dashboard</code>

## Expected Final Result

User is authenticated and redirected to `/dashboard`. Auth cookie/token is set. Welcome message visible. Navigation shows user's name/avatar.

## Post-conditions

- Session exists and is valid
  - Audit log entry created for login event
  - Test cleanup: session will be cleared in test teardown
-

# Test Case: TC-AUTH-002

Field	Value
ID	TC-AUTH-002
Title	Login fails with invalid password — correct error shown
Description	Verifies that an incorrect password returns a generic error without revealing whether the email exists (prevents user enumeration).
Module / Feature	Authentication — Login
Priority	Critical
Type	Negative / Security
Requirement	REQ-AUTH-002, SEC-001
Automation Status	Automated

## Preconditions

- User account with email `qa-test@example.com` exists
- User is not logged in

## Test Data

Field	Value
Email	<code>qa-test@example.com</code>
Password	<code>DefinitelyWrongPassword123!</code>

## Test Steps

Step	Action	Expected Result
1	Navigate to <code>/login</code>	Login form displayed
2	Enter valid email	Email entered
3	Enter wrong password	Password masked
4	Click "Log In"	Form submits
5	Observe response	Error message displayed: "Invalid email or password" (generic, not "wrong password")
6	Verify URL	Still on <code>/login</code> (no redirect)
7	Verify no session	No auth cookie set

## Expected Final Result

Generic error message shown. User remains on login page. No session created. Error does not reveal whether the email exists.

---

## Related Documents

- [Test Strategy](#)
  - [Test Plan](#)
  - [E2E Test Plan](#)
- 

## Approval

Role	Name	Date	Signature
Author			
Reviewer			
Approver			

---

Revision #3

Created 2026-02-24 14:53:49 UTC by John

Updated 2026-05-25 07:34:10 UTC by John