

Testing checklist: LLM-based assistant

Source: <https://howtotestaproduct.com/how-to-test/llm-assistant/>

Generated: 2026-10-07. Licence: CC BY 4.0.

☐ 1. release smoke tests

Done when: On the release build, a signed-in user sends a prompt, gets an answer, triggers one tool call, finds the chat saved in history, and one fixed policy question returns its approved answer.

Risks: dependency-failure, wrong-decisions. Tool class: browser automation. Usually run by: QA engineer.

☐ 2. exploratory testing

Done when: A session log covers policy questions, crisis phrases and off-topic requests, and records each answer next to the approved policy text.

Risks: wrong-decisions, user-harm. Tool class: test management. Usually run by: QA engineer.

☐ 3. compliance checks

Done when: A signed review shows the age gate, the training-data opt-out and the AI disclosure, with a screenshot of each from the release build.

Risks: regulatory-penalty, user-harm. Tool class: test management. Usually run by: product manager.

☐ 4. end-to-end functional tests

Done when: A user who types a self-harm phrase gets the crisis message and helpline link in the same turn, on web and both mobile apps.

Risks: user-harm. Tool class: browser automation. Usually run by: QA engineer.

☐ 5. penetration testing

Done when: A tester planting instructions in an uploaded file or a retrieved page cannot make the assistant call a tool, reveal its system prompt or show another user's data.

Risks: fraud-abuse, data-breach. Tool class: proxy. Usually run by: security specialist.

☐ 6. security scanning

Done when: The scan of the chat API and history store shows no open high-severity finding, and one account's conversation IDs return nothing from another account's session.

Risks: data-breach. Tool class: DAST scanner. Usually run by: security specialist.

☐ 7. performance and load testing

Done when: One account sending prompts in a loop hits its own limit, and other accounts keep getting answers at their usual speed.

Risks: outage-under-load. Tool class: load generator. Usually run by: developer.

☐ 8. resilience and failure testing

Done when: With the model API returning errors or timeouts, the user sees a retry message in the chat and no half-written answer is saved as final.

Risks: dependency-failure. Tool class: chaos tool. Usually run by: developer.

☐ 9. regression suite

Done when: A fixed question set with approved answers is rerun after every prompt, model or retrieval change, and a named owner reviews every changed answer.

Risks: wrong-decisions. Tool class: test management. Usually run by: QA engineer.