



BUILT FOR SHOPIFY

Readiness Guide

A practical field manual for the 77-point quality bar

SEPTEMBER 2026 EDITION

Independent guidance by Stormdev

START HERE

The standard behind the badge

Built for Shopify (BFS) is Shopify's highest app-quality recognition. The program evaluates safety, performance, integration, design, merchant utility, and category-specific use of Shopify primitives.

SCOPE OF THIS EDITION

This guide translates Shopify's 77 published leaf criteria into practical pass conditions and evidence requests. Criteria were checked against Shopify's official requirements and changelog on 11 September 2026. Shopify can change the program; confirm the linked official pages before every application.

What this guide is

- A project-neutral checklist for planning, code review, manual QA, and application evidence.
- A way to distinguish what source code can prove from what requires live-store testing or Partner Dashboard telemetry.
- A category map so you can resolve conditional requirements before implementation and before App Store classification.
- An application gate designed to reduce avoidable review rounds.

What this guide is not

- It is not an approval guarantee, legal opinion, security certification, or replacement for Shopify's current documentation.
- A green local build is not proof of performance, live configuration, clean uninstall, mobile behavior, or Partner standing.
- An item marked not applicable still needs a written applicability reason; silence is not evidence.

OPERATING MODEL

Use one matrix, five statuses

Record every criterion exactly once for one fixed release candidate. Avoid a vague overall percentage: one applicable failure can block the application.

Status	Use it when	Required next step
PASS	Sufficient code, live, or Dashboard evidence supports the criterion.	Link the evidence and keep the release candidate fixed.
FAIL	The product contradicts an applicable requirement or lacks a required primitive.	Fix, retest, and replace stale evidence.
VERIFY	Only telemetry, configuration, or a live walkthrough can decide.	Name the owner, artifact, date, and pass condition.
CONDITIONAL	The criterion becomes binding if Shopify applies the category or capability.	Resolve classification in writing before applying.
N/A	The app does not provide the defined capability.	Write a short product-based rationale.

FREEZE THE CANDIDATE

Record the repository commit, Shopify app version, deployed backend build, scopes, extensions, configuration, app category, test store, and test account. Evidence for one version must not be used to justify another.

Application gate

- All 36 general criteria pass or have authoritative passing evidence.
- Every relevant category is named; every applicable category criterion passes.
- Distribution metrics and required 28-day sample counts are green in Partner Dashboard.
- A clean store completes install, onboarding, core workflow, billing, error recovery, mobile use, and uninstall.
- The evidence packet points to the exact released candidate and has no expired credentials or stale recordings.

READINESS MAP

The 77 criteria at a glance

The program combines 36 general criteria with 41 category-specific criteria. Category count does not mean all 41 apply to every app.

Part	Area	Count	Primary evidence
1	Prerequisites	5	Partner Dashboard, App Store baseline
2	Performance	5	28-day telemetry and Shopify assessments
3	Integration	7	Released source, configuration, live workflows
4	Design	19	Desktop/mobile walkthroughs and interaction tests
5	Category-specific	41	Category primitives, extensions, APIs, operational metrics
TOTAL	Published leaf criteria	77	One evidence row per criterion

Recommended review order

- 1. Resolve category and App Store classification first.
- 2. Run the source-level safety, privacy, integration, and release audit.
- 3. Fix design and workflow failures in one coherent UI pass.
- 4. Release the candidate and collect live, mobile, and telemetry evidence.
- 5. Apply only after a final independent matrix review finds no unresolved failure or verify row.

REVIEW RISK

If the same BFS criterion fails three application rounds in a row, Shopify states that the application is suspended for three months. Approved apps are reviewed annually; failures during annual review generally have a 60-day remediation window.

PART I

Universal requirements

These 36 criteria apply broadly. Criterion 2.3.1 applies specifically where carrier rates affect checkout.

1. Prerequisites

1.1.1

Meet App Store requirements

PASS WHEN: Remain compliant with the entire current App Store rule set, not only the BFS page.

VERIFY WITH: **Current listing, review-ready app, policy/security/privacy/billing evidence.**

1.1.2

Good Partner standing

PASS WHEN: No active or outstanding Partner Account infractions; prior enforcement may still affect eligibility.

VERIFY WITH: **Partner Dashboard standing and enforcement history.**

1.2.1

Minimum installs

PASS WHEN: At least 50 net installs from active shops on paid Shopify plans.

VERIFY WITH: **BFS Distribution card in Partner Dashboard.**

1.2.2

Minimum reviews

PASS WHEN: At least five Shopify App Store reviews.

VERIFY WITH: **App listing and Partner Dashboard.**

1.2.3

Minimum recent rating

PASS WHEN: Meet Shopify's unpublished recent-rating threshold.

VERIFY WITH: **Partner Dashboard rating indicator; public lifetime rating is not enough.**

2. Performance

2.1.1

Admin LCP

PASS WHEN: Largest Contentful Paint at p75 is 2.5 seconds or less, with at least 100 calls in 28 days.

VERIFY WITH: **28-day Web Vitals card with p75 and sample count.**

2.1.2

Admin CLS

PASS WHEN: Cumulative Layout Shift at p75 is 0.1 or less, with at least 100 calls in 28 days.

VERIFY WITH: **28-day Web Vitals card with p75 and sample count.**

2.1.3

Admin INP

PASS WHEN: Interaction to Next Paint at p75 is 200 ms or less, with at least 100 calls in 28 days.

VERIFY WITH: **28-day Web Vitals card with p75 and sample count.**

2.2.1**Storefront impact**

PASS WHEN: The app reduces the storefront Lighthouse performance score by no more than 10 points.

VERIFY WITH: **Shopify before/after storefront performance assessment.**

2.3.1**Checkout speed**

PASS WHEN: For carrier-rate apps: at least 1,000 requests in 28 days, p95 at 500 ms or less, failure rate at 0.1% or less.

VERIFY WITH: **Partner Dashboard checkout-performance telemetry.**

3. Integration and installation

3.1.1**Embed in Shopify Admin**

PASS WHEN: Use the latest App Bridge, load app-bridge.js in every app document, and use ID-token authentication.

VERIFY WITH: **Released config, document head, auth path, embedded live walkthrough.**

3.1.2**Keep primary workflows in Shopify**

PASS WHEN: Merchants can complete core work in Admin unless a documented exception genuinely requires a standalone surface.

VERIFY WITH: **Workflow map and clean-store recording.**

3.1.3**Seamless Shopify sign-up**

PASS WHEN: No unnecessary second account or sign-up gate after install; supported B2B exceptions still begin in Admin.

VERIFY WITH: **Fresh-install walkthrough using Shopify identity.**

3.1.4**Simplified monitoring in Admin**

PASS WHEN: Useful key metrics and status are available on the app home even if detailed reporting lives elsewhere.

VERIFY WITH: **Home-page screenshots populated with realistic data.**

3.1.5**Connection settings in Admin**

PASS WHEN: Merchants can connect, configure, disconnect, and understand third-party integrations inside the embedded app.

VERIFY WITH: **Connection lifecycle walkthrough, including disconnect.**

3.2.1**Clean uninstallation**

PASS WHEN: Use theme app extensions for storefront elements and remove or durably process app-owned data/artifacts when uninstalled.

VERIFY WITH: **Uninstall test with expired/revoked tokens and deletion proof.**

3.2.2**Do not mutate theme files**

PASS WHEN: Do not use the Asset API to add, edit, or delete merchant theme files unless a listed exception applies.

VERIFY WITH: **Scope review, source search, theme diff before/after install.**

4.1 Design: familiar

4.1.1	Follow Shopify UX best practices PASS WHEN: UI feels native to Admin: familiar cards, spacing, buttons, typography, contrast, hierarchy, and stable loading. VERIFY WITH: Desktop/mobile UI inventory against Shopify patterns.
4.1.2	Mobile-friendly PASS WHEN: No page-wide horizontal scroll, inaccessible content, or desktop columns squeezed into unusable mobile layouts. VERIFY WITH: 375 px, tablet, Shopify Mobile, and zoom walkthroughs.
4.1.3	Concise app name PASS WHEN: The pinned app name remains fully visible in Shopify Admin without truncation. VERIFY WITH: Pinned navigation screenshot on desktop.
4.1.4	Use App Bridge navigation PASS WHEN: Use s-app-nav; highlight the correct parent, avoid duplicate Home items and emojis. VERIFY WITH: Navigation source plus route-by-route walkthrough.
4.1.5	Use the Contextual Save Bar PASS WHEN: Substantial forms expose save/discard and block navigation while unsaved changes exist. VERIFY WITH: Dirty, save, discard, failure, and leave-page tests.
4.1.6	Use modals appropriately PASS WHEN: Use s-modal headings and action slots; preserve focus, Escape, close, and focus restoration behavior. VERIFY WITH: Keyboard-only modal recording and DOM/accessibility check.

4.2 Design: helpful

4.2.1	Clear language PASS WHEN: Headings, labels, actions, units, spelling, and grammar are concise and understandable in every locale. VERIFY WITH: UI copy inventory and localization review.
4.2.2	Helpful onboarding PASS WHEN: Onboarding is concise, findable, removable after completion, and gets the merchant to core value without unrelated installs. VERIFY WITH: Fresh-store onboarding recording and completion state.
4.2.3	Helpful app home PASS WHEN: The home page surfaces relevant status, setup progress, issues, and useful next actions rather than a static link hub. VERIFY WITH: Populated and empty-state home screenshots.

4.2.4

Helpful error messages

PASS WHEN: Errors are red, contextual where possible, actionable, and remain until fixed or explicitly dismissed.

VERIFY WITH: Field, page, network, permission, and retry-state recording.

4.2.5

Guide logical actions

PASS WHEN: The safest and most likely next action is visually dominant; destructive alternatives are clearly secondary.

VERIFY WITH: Action-group review across screens and dialogs.

4.2.6

Visible previews

PASS WHEN: Visual customization updates a live preview; on desktop controls and preview are visible together.

VERIFY WITH: Editor recording covering success, error, and mobile layouts.

4.3 Design: user-friendly

4.3.1

No false claims

PASS WHEN: Do not guarantee outcomes or present success before work is durably accepted.

VERIFY WITH: Claim inventory across UI, onboarding, emails, and listing.

4.3.2

Do not pressure merchants

PASS WHEN: No countdown pressure, guilt/shame copy, or incentives tied to five-star reviews.

VERIFY WITH: Promotion, trial, review-request, and cancellation copy review.

4.3.3

Do not distract

PASS WHEN: Avoid unsolicited modals, popovers, strong colors, and attention animations unrelated to merchant actions.

VERIFY WITH: Timed/page-load behavior recording; reduced-motion test.

4.3.4

Do not overwhelm

PASS WHEN: Break up complex forms, keep copy scannable, and avoid multiple nearby banners.

VERIFY WITH: Information-density and banner inventory.

4.3.5

Do not impersonate Shopify

PASS WHEN: Branding cannot be mistaken for Shopify, a first-party app, or Sidekick.

VERIFY WITH: Icon, name, palette, AI treatment, and listing review.

4.3.6

Dismissible ads

PASS WHEN: Promotions can be dismissed and stay dismissed for a sensible period.

VERIFY WITH: Dismiss, reload, navigation, and re-login tests.

4.3.7

Label and disable premium features

PASS WHEN: Gated controls look and behave disabled, identify the required tier, and hide Plus-only features from non-Plus stores.

VERIFY WITH: Plan matrix across eligible and ineligible shops.

PART II

Category-specific requirements

Classify by actual functionality, not by what the team hopes is non-applicable. An app can belong to more than one category. If a public listing category conflicts with product behavior, resolve it before applying.

CATEGORY DECISION RECORD

For every category below, write Applicable, Not applicable, or Needs Shopify confirmation. Include one sentence describing the relevant product capability and one link to evidence. Treat ambiguous analytics, affiliate, forms, discount, and marketing behavior as conditional until resolved.

Ads apps

Apps that create or manage digital advertising campaigns.

5.1.1**Use Web Pixels**

PASS WHEN: Use Web Pixel extensions for relevant attribution, audience, analytics, pixel, retargeting, or lookalike events; no ScriptTags or copied JavaScript.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.1.2**Use Shopify Segments**

PASS WHEN: Allow any Shopify segment for multi-customer targeting and expose the operation through a customer segment action extension.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

Affiliate program apps

Apps that manage commission-based influencer or affiliate promotion.

5.2.1**Use Web Pixels**

PASS WHEN: Use Web Pixel extensions for relevant Shopify events; no ScriptTags or merchant-copied JavaScript.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

Analytics apps

Apps that provide data-driven insight into store performance.

5.3.1**Use Web Pixels**

PASS WHEN: Use Web Pixel extensions for relevant Shopify events; no ScriptTags or merchant-copied JavaScript.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

Carrier services apps

Apps that provide real-time carrier-calculated rates.

5.4.1

Respond quickly

PASS WHEN: At least 1,000 requests in 28 days; fewer than 500 ms for 95% of rate calls.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.4.2

Respond reliably

PASS WHEN: Successfully respond to 99.9% of rate requests over 28 days.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

Discount apps

Apps that define or configure price reductions.

5.5.1

Use discount primitives

PASS WHEN: Use Shopify Functions for custom discount types or native discount APIs.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.5.2

No draft-order custom discounts

PASS WHEN: Do not automate customer-driven discounts by creating draft orders with custom discounts.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.5.3

Bulk codes under one discount

PASS WHEN: Use discountRedeemCodeBulkAdd when one discount needs multiple redeem codes.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.5.4

High-quality Create discount links

PASS WHEN: Admin Create discount links must open the matching creation workflow in the embedded app.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

Email marketing apps

Apps for targeted email campaigns.

5.6.1

Use Web Pixels

PASS WHEN: Use Web Pixel extensions when automation, segmentation, analytics, or pixels need Shopify events.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.6.2

Sync customer data

PASS WHEN: Synchronize relevant customer data to and from Shopify in line with API terms.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.6.3

Use Shopify Segments

PASS WHEN: Support any Shopify segment and expose multi-customer operations through a segment action.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.6.4

Use the Visitors API

PASS WHEN: Log permitted identifiers such as email or phone when storefront visitors provide them.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

Forms apps

Apps that collect custom customer fields, preferences, or inquiries.

5.7.1

Use Shopify Segments

PASS WHEN: Support any Shopify segment and expose relevant multi-customer operations through a segment action.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.7.2

Use the Visitors API

PASS WHEN: Log permitted identifiers such as email or phone when storefront visitors provide them.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.7.3

Sync customer data

PASS WHEN: Synchronize relevant customer data to and from Shopify with a defined ownership/conflict model.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

Fulfillment services apps

Apps using their own location to prepare and ship merchant orders.

5.8.1

Actively fulfill

PASS WHEN: Complete at least 100 fulfillment orders in the last 28 days.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.8.2

Complete assigned orders

PASS WHEN: Complete 97% of assigned orders in the 28-day window, excluding orders created in the last seven days.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.8.3

Callback success

PASS WHEN: Successfully respond to 99% of Shopify callback requests in the last 28 days.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.8.4

Wait for merchant request

PASS WHEN: Only fulfill after the merchant requests fulfillment.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.8.5

Add tracking promptly

PASS WHEN: Add tracking to 80% of created fulfillments within one hour.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.8.6

Respond to fulfillment requests

PASS WHEN: Accept or reject 95% of fulfillment requests within 24 hours.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.8.7

Respond to cancellations

PASS WHEN: Accept or reject 99% of cancellation requests within 24 hours.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

Invoices and receipts apps

Apps that generate invoices or packing slips.

5.9.1

Print from Orders

PASS WHEN: Provide an admin print action for one order and selected orders on the index page.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

Product bundles apps

Apps that group products for sale as one unit.

5.10.1

Use bundle primitives

PASS WHEN: Create static bundles with Admin GraphQL or customized bundles with a cartTransform function, except documented unsupported cases.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

Product reviews apps

Apps that collect product reviews.

5.11.1

Provide a Flow trigger

PASS WHEN: Start a Shopify Flow workflow whenever a new review is collected.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

5.11.2

Use customer admin blocks

PASS WHEN: Show a customer's submitted reviews in an admin block on customer detail pages.

VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

Returns and exchanges apps

Apps that manage returns, exchanges, or refunds.

5.12.1	Sync return lifecycle PASS WHEN: Communicate creation, shipping, restocking, line removal, cancellation, closure, and refunds through the appropriate APIs. VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.
5.12.2	Include exchange line items PASS WHEN: Create and remove exchange line items on the order as the exchange changes. VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.
5.12.3	Include fees PASS WHEN: Add applicable shipping and restocking fees on the order. VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.
5.12.4	Use Customer Account API auth PASS WHEN: For buyer self-service, use Customer Account API as primary authentication. Effective December 1, 2026. VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

SMS marketing apps

Apps for targeted SMS campaigns.

5.13.1	Use Web Pixels PASS WHEN: Use Web Pixel extensions when automation, segmentation, analytics, or pixels need Shopify events. VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.
5.13.2	Sync customer data PASS WHEN: Synchronize relevant customer data to and from Shopify in line with API terms. VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.
5.13.3	Use Shopify Segments PASS WHEN: Support any Shopify segment and expose multi-customer operations through a segment action. VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.
5.13.4	Use the Visitors API PASS WHEN: Log permitted identifiers such as email or phone when storefront visitors provide them. VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

Subscription apps

Apps that enable recurring customer product purchases.

5.14.1	Use subscription objects and APIs PASS WHEN: Use Selling Plans, Subscription Contracts, and Customer Payment Methods for recurring purchases. VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.
5.14.2	Use theme app blocks PASS WHEN: Add purchase options to product pages with Online Store 2.0-compatible app blocks. VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.
5.14.3	Follow subscription UX PASS WHEN: Clearly show plan name, price, and savings on product, cart, and order pages; match the theme by default. VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.
5.14.4	Use Customer Account UI extensions PASS WHEN: Let customers view and manage subscriptions through Customer Account UI extensions. VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.
5.14.5	Use Customer Account API auth PASS WHEN: For buyer self-service, use Customer Account API as primary authentication. Effective December 1, 2026. VERIFY WITH: Released primitive or extension, clean-store walkthrough, and category telemetry where required.

IMPORTANT DEADLINE

Customer Account authentication

Shopify has published new authentication requirements for returns/exchanges and subscription apps with buyer-facing self-service experiences.

EFFECTIVE 1 DECEMBER 2026

Affected apps must use the Customer Account API as the primary method for customer authentication. The current requirements page already displays criteria 5.12.4 and 5.14.5; plan and test the migration before the deadline.

Acceptance evidence

- A buyer can enter the self-service flow through Shopify customer accounts without a second credential system.
- Fresh, returning, logged-out, expired-session, and account-switch flows behave correctly.
- Authorization is enforced server-side for every return, exchange, and subscription object.
- Customer Account UI extensions expose the required management tasks without sending buyers to an avoidable external login.
- The App Store listing and review instructions accurately describe supported account requirements.

Fulfillment thresholds changed in August 2026

Shopify adjusted criteria 5.8.2, 5.8.6, and 5.8.7. Use the current thresholds in this guide and the live requirements page; do not reuse older matrices or screenshots.

PART III

The audit beneath criterion 1.1.1

BFS re-audits App Store compliance. A polished interface can still fail if authorization, privacy, billing, reliability, or core-workflow integrity is weak.

AUDIT	Authentication and tenancy PASS WHEN: Every app route, API, job, webhook, and object lookup proves the requesting shop or buyer can access the target resource. PROVE WITH: Source path, automated test, negative-case recording, and released-version proof.
AUDIT	Webhook trust PASS WHEN: Verify raw-body HMAC before processing. Handle duplicates, stale events, retries, and out-of-order delivery safely. PROVE WITH: Source path, automated test, negative-case recording, and released-version proof.
AUDIT	Input authorization PASS WHEN: Validate identifiers against the merchant-configured allowlist or server-side source of truth; do not trust hidden fields or client state. PROVE WITH: Source path, automated test, negative-case recording, and released-version proof.
AUDIT	Injection and serialization PASS WHEN: Escape HTML, attributes, URLs, CSV, logs, and inline script JSON. Test script-closing strings and Unicode edge cases. PROVE WITH: Source path, automated test, negative-case recording, and released-version proof.
AUDIT	Durable success PASS WHEN: Show success only after the request is durably accepted. Background work needs a recoverable receipt, idempotency key, and status model. PROVE WITH: Source path, automated test, negative-case recording, and released-version proof.
AUDIT	Concurrency PASS WHEN: Reserve quotas and uniqueness transactionally. Test two last-slot submissions, retries, double-clicks, and webhook replays. PROVE WITH: Source path, automated test, negative-case recording, and released-version proof.
AUDIT	Privacy and redaction PASS WHEN: Erase or irreversibly unlink direct and derived identifiers across primary data, logs, media, processors, and grouping tables. PROVE WITH: Source path, automated test, negative-case recording, and released-version proof.

AUDIT	Clean uninstall PASS WHEN: Do not require a refreshable Admin token to accept a valid uninstall event. Delete or durably queue tenant cleanup. PROVE WITH: Source path, automated test, negative-case recording, and released-version proof.
AUDIT	Consent enforcement PASS WHEN: Server-side rules must enforce required consent and preserve consent provenance; browser validation is only usability. PROVE WITH: Source path, automated test, negative-case recording, and released-version proof.
AUDIT	Billing integrity PASS WHEN: Use Shopify billing, handle accept/decline/cancel, upgrades/downgrades, reinstall, trials, and webhook races. PROVE WITH: Source path, automated test, negative-case recording, and released-version proof.
AUDIT	Scopes and API use PASS WHEN: Request only necessary scopes, prefer optional scopes for non-universal features, and use current GraphQL/platform primitives. PROVE WITH: Source path, automated test, negative-case recording, and released-version proof.
AUDIT	Release reproducibility PASS WHEN: Lock dependencies, gate release on checks, map commit to Shopify version and backend deploy, and remove placeholder configuration. PROVE WITH: Source path, automated test, negative-case recording, and released-version proof.

APP STORE BASELINE

High-risk areas to review

Criterion 1.1.1 points to the full Shopify App Store requirements, which are broader and subject to change. Use the official page as the controlling source.

- Embedded authentication uses session/ID tokens and works without third-party cookies or browser local-storage assumptions.
- Transactions use Shopify checkout and approved platform APIs; payment, refund, marketplace, lending, and POS restrictions are respected.
- App pricing uses Shopify App Pricing or Billing API and supports plan changes without support intervention or reinstall.
- Core workflows are reliable, synchronized data remains accurate, and UI/API errors do not block the review.
- New public apps use GraphQL Admin API; legacy REST use is identified and removed where required.
- Installation starts on Shopify, OAuth happens immediately, and reinstall authenticates again before UI access.
- TLS is valid, access scopes are necessary, privacy webhooks are correct, and protected customer data rules are followed.
- Listing name, tags, categories, pricing, claims, languages, geography, screenshots, and supported features match the product.
- Review incentives are neutral; no feature, reward, or discount depends on a positive or five-star review.
- Test credentials, demo screencast, setup steps, emergency contact, and reviewer access remain current.

DO NOT TURN THIS PAGE INTO A STATIC CHECKLIST

Shopify explicitly says App Store requirements can change. Re-open the official page during the final release review and record the checked date in the evidence matrix.

PART IV

Evidence packet

A useful evidence packet lets a reviewer reproduce the app's intended behavior without guessing which version, store, plan, or category they are seeing.

Owner	Artifact	Pass condition
Partner admin	Distribution export	Standing, installs, reviews, rating, and performance indicators pass.
Release owner	Candidate manifest	Commit, backend, app version, scopes, extensions, and config match.
QA	Desktop/mobile walkthrough	No clipping, inaccessible content, broken navigation, or lost changes.
QA	Error-state recording	Errors persist, explain recovery, and sit next to fields where possible.
Security	Negative test suite	Cross-tenant IDs, replay, injection, race, and forged client state are safe.
Privacy	Redaction/uninstall fixture	Identity cannot be reconstructed; cleanup succeeds without Admin token refresh.
Product	Category decision	Every category is applicable, N/A with rationale, or confirmed by Shopify.
Performance	28-day metrics	Threshold and sample count pass for every applicable performance criterion.

Evidence hygiene

- Name files with criterion ID, candidate version, store, and date.
- Capture the full browser/Admin context; do not crop away store, plan, route, or measurement period.
- Prefer short recordings for interactive requirements and screenshots/exports for Dashboard facts.
- Link every matrix row to one best artifact, then add supporting evidence only when it resolves ambiguity.
- Retire evidence after a material change to UI, scopes, extension set, category, billing, or core workflow.

MANUAL QA

The reviewer journey

Run this on a clean development store and again on the exact submission candidate. Include desktop and Shopify Mobile where supported.

QA	1. Install PASS WHEN: Start from a Shopify-owned surface, complete OAuth, land in the app, and confirm the correct shop/plan context. CAPTURE: Recording plus defect-free result linked to the frozen candidate.
QA	2. Onboard PASS WHEN: Finish setup without outside instructions; remove completed onboarding UI; verify empty, partial, and already-configured states. CAPTURE: Recording plus defect-free result linked to the frozen candidate.
QA	3. Core value PASS WHEN: Complete each listing-promised workflow using realistic data, third-party connection states, and applicable Shopify primitives. CAPTURE: Recording plus defect-free result linked to the frozen candidate.
QA	4. Edit safely PASS WHEN: Trigger dirty state, save, discard, navigation blocking, validation, network failure, permission failure, and retry. CAPTURE: Recording plus defect-free result linked to the frozen candidate.
QA	5. Responsive and accessible PASS WHEN: Use keyboard only, 200% zoom, narrow mobile, long translated copy, focus restoration, and reduced motion. CAPTURE: Recording plus defect-free result linked to the frozen candidate.
QA	6. Billing and gating PASS WHEN: Test free, trial, paid, upgrade, downgrade, cancel, declined approval, Plus-only, and ineligible-shop states. CAPTURE: Recording plus defect-free result linked to the frozen candidate.
QA	7. Storefront/buyer flow PASS WHEN: Test supported themes, app blocks, consent, identification, customer accounts, checkout, and failure recovery. CAPTURE: Recording plus defect-free result linked to the frozen candidate.
QA	8. Uninstall and reinstall PASS WHEN: Uninstall with expired/revoked credentials, prove cleanup, confirm no theme residue, then reinstall through OAuth. CAPTURE: Recording plus defect-free result linked to the frozen candidate.

SUBMISSION

Final 48-hour runbook

Stop feature work. Treat the application candidate as a release, not as a moving development branch.

T-48 hours

- Freeze commit, Shopify app version, backend deployment, scopes, extensions, category, pricing, and listing copy.
- Re-run automated tests and the source-level risk review; close or explicitly accept every release blocker.
- Confirm the Distribution page, Web Vitals sample counts, storefront impact, and category telemetry.

T-24 hours

- Run the clean-store reviewer journey on desktop and mobile; refresh every changed screenshot or recording.
- Test test-user credentials, external connections, billing approval, sample data, and reviewer instructions from scratch.
- Review UI and listing claims together; qualify outcomes and remove stale feature, language, plan, or geography claims.

Submit

- Have a team member with Manage apps permission apply from Apps > app > Distribution > Apply now.
- Keep monitoring, support, and emergency contacts staffed; preserve logs for the exact review window.
- Respond to each reviewer issue with the criterion, root cause, released fix, and reproducible evidence.

SHIP THE PROOF, NOT JUST THE FIX

A corrected implementation without updated live evidence can remain a Verify row. Re-record the exact behavior after deployment.

SOURCES

Official references

The guide uses Shopify's official developer documentation as the controlling source. The checklist language is summarized for practical use; the linked pages remain authoritative.

Built for Shopify requirements

<https://shopify.dev/docs/apps/launch/built-for-shopify/requirements>

Built for Shopify overview and application process

<https://shopify.dev/docs/apps/launch/built-for-shopify>

Built for Shopify annual reviews

<https://shopify.dev/docs/apps/launch/built-for-shopify/annual-reviews>

Shopify App Store requirements

<https://shopify.dev/docs/apps/launch/shopify-app-store/app-store-requirements>

August 2026 fulfillment-services update

<https://shopify.dev/changelog/updated-built-for-shopify-requirements-for-fulfillment-services-apps>

December 2026 Customer Account API deadline

<https://shopify.dev/changelog/built-for-shopify-requirements-for-returns-and-exchanges-and-subscription-apps>

CHECKED 11 SEPTEMBER 2026

Before using this guide for a future application, review Shopify's requirements page and Built for Shopify changelog for additions, threshold changes, grace periods, and effective dates.

Independent notice

Built for Shopify is a Shopify program. This independent guide is not endorsed by Shopify and does not guarantee acceptance. Product, legal, privacy, and security decisions remain the app developer's responsibility.

STORMDEV