

Make a vibe-coded app production ready

<https://prompt-to-prototype.com/lifecycle/production-ready/> · as of 2026-10-22

A vibe-coded app is production ready when it has an accountable owner, company sign-in, tested data rules, recorded checks and a support path, with the depth set by its [risk tier](#). Before other people rely on it, it's registered with a named owner, runs on approved hosting with company sign-in, keeps no keys, passwords or tokens in code, front end or prompts (secrets are held server-side, in a secrets manager or the platform's server-only secret store), and has a recorded [V.E.R.I.F.Y. self-check](#), a one-page app record and a support path. From Important up it also needs data-owner sign-off for Confidential or Restricted data, negative access tests, a technical check, a runbook and a named backup; the checklist below, as of October 2026, marks each line as a gate or good practice.

Who this is for

Builders and app owners preparing to go live, and the reviewers who sign off. You get one checklist with a tier column and a "Who checks" column, a downloadable copy, the accessibility duties, and what to do when your builder can't do something. Reading time: about 10 minutes.

Not legal advice

Not legal advice. This page summarizes laws as of October 2026 for a general audience. It was written with an AI legal-research persona and hasn't been reviewed by a lawyer for your situation. Ask your counsel.

Jump to: [the checklist](#) · [common blockers](#) · [if your builder can't do this](#) · [accessibility](#) · [sign-off](#)

What does a vibe-coded app need before go-live?

This is the go-live list the [deploy gate](#) points to. Download it as [go-live-checklist.csv](#) (Prompt to Prototype, Make a vibe-coded app production ready, <https://prompt-to-prototype.com/lifecycle/production-ready/>, as of 2026-10-21. Licensed CC BY-SA 4.0). For how to check each security line, use the [vibe coding security checklist \(CSV\)](#). Using or adapting either file inside your business isn't distribution under the licence.

- **Kind.** A **Gate** line is one cell of the [gate matrix](#), named by stage and gate. If this list and the matrix differ, the matrix wins. A *Good practice* line can be skipped with a reason written in the app record; that needs no exception.
- **From.** The lowest tier the line applies to, and every tier above. "Every tier" includes Sandbox; "Team only" stops applying at Important.
- **Who checks.** The builder can't tick a line that names someone else, and never approves their own exception ([the roles](#)).
- **No technical reviewer?** The app stays at Team scope until the technical check is done. Until your published list of technical reviewers exists, the IT/platform owner names them.

Ownership

Done	Item	Kind	From	Who checks
☐	In the app registry , with an app record naming the app owner , tier and data class	Gate (Intake: Register)	Team	Builder; enablement lead at triage
☐	The six tier questions and both flags answered and recorded	Gate (Intake: Answer the six questions and the flags)	Team	Builder; enablement lead at triage
☐	Kill criteria and a retire-by date set	Gate (Intake: Kill criteria and retire-by date)	Team	App owner
☐	Data-owner sign-off, if the app holds Confidential or Restricted data. No exception is possible	Gate (Intake: Register)	Important	Data owner
☐	Consent gate or jurisdiction flag yes: counsel answered before collecting or building. No exception is possible	Gate (Intake: Counsel gates)	Team	Counsel, through the counsel route
☐	Audit-scope flag yes: the compliance owner's change-management and evidence requirements are set and met. They apply at Team exactly as at higher tiers; the flag sets no minimum tier	Gate (Intake: Audit-scope flag; Change: Audit scope)	Team	Compliance owner
☐	A graduation plan agreed, and IT or engineering recorded as co-owner	Gate (Intake: Register; Operate: Graduation)	Critical	IT/platform owner
☐	Next scheduled review date set, 12 months out	Gate (Operate: Scheduled review)	Team only	App owner
☐	Next scheduled review date set, 6 months out	Gate (Operate: Scheduled review)	Important	App owner
☐	A quarterly access review booked, as well as the 6-monthly review	Gate (Operate: Scheduled review)	Critical	App owner, with IT
☐	Any downgrade or bounded exception in force is in the app record with its approver, reason and review or expiry date	Gate (Operate: Downgrade in force; Exception in force)	Team	Enablement lead (Critical: IT/platform owner and security reviewer)
☐	A plan for when the builder leaves	<i>Good practice</i>	Team	App owner

Access

Done	Item	Kind	From	Who checks
☐	Built and hosted with approved tools on business accounts, never a personal account (never-do 3)	Gate (Build: Tools and accounts)	Every tier	Builder
☐	On an approved hosting pattern with company sign-in (SSO); shared by group, not "anyone with the link" (access)	Gate (Deploy: Hosting)	Team	Builder; technical reviewer from Important

Done	Item	Kind	From	Who checks
☐	Not reachable from the internet without company sign-in (Floor 2); a Team app that needs internet reach is re-tiered first	Gate (Deploy: Internet reach (Floor 2))	Team only	Builder, by the signed-out test; IT's scheduled exposure scan, where it covers Team apps
☐	Internet reach without company sign-in only on an approved public-facing pattern	Gate (Deploy: Internet reach (Floor 2))	Important	Technical reviewer
☐	No floor breach open; a Floor 2 fix is confirmed by a clean exposure scan (floor-breach rule)	Gate (Operate: Floor breach)	Every tier	Technical reviewer or IT
☐	Runs on IT's standard platform	Gate (Build: Tools and accounts; Deploy: Hosting)	Critical	IT/platform owner

Data

Done	Item	Kind	From	Who checks
☐	Data goes only to tools and AI services approved for its data class (never-do 1, the M1 approval gate). No exception is possible	Gate (Build: Floors and never-dos; Build: AI feature (M1))	Every tier	Builder
☐	Data rules deny by default on every table and storage bucket (data rules)	Gate (Deploy: Hosting)	Important	Technical reviewer
☐	Negative access tests pass: a signed-out visitor and a lowest-role test account see nothing they shouldn't	Gate (Verify: Tests)	Important	Technical reviewer
☐	The signed-out and second-account tests in how to test your app pass	<i>Good practice</i>	Team	Builder

Security

Most of these lines are enforced for you by the [deployment guardrails](#); each links to its checks. Don't assume the vendor secured it: builder vendors have called exposed data "by design" and told customers the configuration was theirs to fix (AP3).¹²

Done	Item	Kind	From	Who checks
☐	No keys, passwords or tokens in code, front end or prompts; secrets are held server-side, in a secrets manager or the platform's server-only secret store (never-do 2; where API keys and secrets leak)	Gate (Build: Floors and never-dos)	Every tier	Builder; technical reviewer from Important
☐	No agent or AI feature combines private data, untrusted content and a way to send data out (Floor 1, the trifecta rule)	Gate (Build: Floors and never-dos)	Every tier	Builder
☐	Coding agents run without permission-bypass modes (never-do 5)	Gate (Build: Floors and never-dos)	Every tier	Builder

Done	Item	Kind	From	Who checks
☐	An agent with write access to live systems had a sandbox run first, has a scoped credential and a restore you've tested (never-do 4)	Gate (Build: Floors and never-dos)	Every tier	Builder; technical reviewer from Important
☐	An AI feature has an eval set with a pass threshold, an AI-feature review, the provider's data terms and a disclosure check (evaluate an AI feature); M2 makes the app Critical	Gate (Build: AI feature (M1))	Team	Builder; technical reviewer sees the eval results from Important
☐	An agent or automation is answered for every granted permission, with its identity, grants and tokens recorded under the app owner's role	Gate (Build: Agents and automations (M3))	Team	App owner
☐	The technical check is recorded (<code>technical_check_on</code>): exposure scan, negative tests and the 30–45 minute fixed-list check (technical check ; what the scan must check)	Gate (Verify: Review)	Important	Technical reviewer
☐	The security review is done	Gate (Verify: Review)	Critical	Security reviewer
☐	The dependency and input-handling checks on the security checklist	<i>Good practice</i>	Team	Builder; technical reviewer from Important

Don't: give an agent standing write access to production

Agents with standing production access have deleted live data (AP4; see [failure patterns](#)). Never-do 4 applies at every tier: a sandbox run first, a scoped credential, and a restore you've tested.

Testing

Done	Item	Kind	From	Who checks
☐	The builder's own test cases pass before every release (test your app)	Gate (Verify: Tests)	Team	Builder
☐	V.E.R.I.F.Y. self-check recorded (the V.E.R.I.F.Y. self-check)	Gate (Verify: Review)	Team	Builder
☐	A business review recorded (business review and technical check)	Gate (Verify: Review)	Important	A colleague who knows the process, not the builder
☐	Independent tests pass	Gate (Verify: Tests)	Critical	IT or engineering, not the builder

Environments

Done	Item	Kind	From	Who checks
☐	Separate test and production, with different access and no live data in test	Gate (Deploy: Hosting)	Important	Technical reviewer
☐	Changes are tried somewhere that isn't live data	<i>Good practice</i>	Team	Builder

Done	Item	Kind	From	Who checks
☐	Production isn't changed by hand in the builder's editor without a recorded change	<i>Good practice</i>	Important	App owner

Operations

Done	Item	Kind	From	Who checks
☐	The platform's audit log is on and kept (logging)	<i>Good practice</i>	Team	Builder
☐	Backups or exports are scheduled	<i>Good practice</i>	Team	Builder
☐	A restore tested by someone other than the builder, dated in the record (<code>last_restore_test_on</code>)	<i>Good practice</i>	Important	Named backup
☐	Someone is alerted if the app stops or its AI spend jumps (with no pipeline: an uptime check in IT's monitoring tool, or a scheduled test sign-in, alerting the registry's support route)	<i>Good practice</i>	Important	App owner, with IT

Accessibility items

Done	Item	Kind	From	Who checks
☐	If people outside the business use the app (Q2(c)) or receive messages it sends or drafts: an accessibility check (WCAG 2.2 AA) of its core tasks and an AI-disclosure check , both recorded	Gate (Verify: Review)	Important	App owner signs; counsel on what law applies (accessibility)
☐	WCAG 2.2 AA checks on the core tasks of an app only staff use	<i>Good practice</i>	Important	Builder
☐	Basic checks: keyboard use, labels, contrast	<i>Good practice</i>	Team	Builder
☐	A named route to ask for an accessible alternative	<i>Good practice</i>	Team	App owner

Documentation

Done	Item	Kind	From	Who checks
☐	A one-page app record	Gate (Deploy: Documentation)	Team	App owner
☐	A runbook: restart, restore, roll back, who to call (document your app)	Gate (Deploy: Documentation)	Important	Named backup, by running the app from it
☐	IT's standard documentation	Gate (Deploy: Documentation)	Critical	IT

Support

Done	Item	Kind	From	Who checks
☐	Team-supported: the builder plus a champion; users know where to report a problem, and the report reaches a person	Gate (Operate: Support level)	Team	App owner
☐	Assisted: a named backup who has run the app, help-desk routing from the registry, and a runbook (support model)	Gate (Operate: Support level)	Important	Enablement lead
☐	IT-supported at normal service levels; the help desk routes reports to IT's queue	Gate (Operate: Support level)	Critical	IT

Rollback

Done	Item	Kind	From	Who checks
☐	Each release passes the tier's tests and review, in version history you can roll back	Gate (Change: Checks per release)	Team	Builder
☐	Releases go through IT's change process	Gate (Change: Checks per release)	Critical	IT
☐	You've rolled back once, so you know it works	<i>Good practice</i>	Team	Builder
☐	The runbook has a written rollback step, including what to do with data written since	<i>Good practice</i>	Important	App owner

What blocks most go-lives, and how do you fix it?

Blocker	Fix
The builder can't use company sign-in	Don't go live; use a different approved tool or hosting pattern (below)
Data is readable without sign-in	Turn on deny-by-default rules, rerun the negative test, then rerun the exposure scan. A clean <i>passive</i> scan isn't proof: it crawls for about a minute and "doesn't perform any actual attacks", ³ while the largest builder exposures were found by pulling public keys from front-end code and querying the backend (AP1, AP2). ⁴ Run the active checks
Keys are in front-end code	Move them server-side (a secrets manager or the platform's server-only secret store) and rotate them: exposed keys are leaked keys
Nobody else has run the app	The named backup runs it from the runbook before go-live
No restore has been tested	Restore a backup into a test copy and check the data
The accessibility check finds many issues	Fix the core task first; record the rest with dates in the app record

Blocker	Fix
The technical-check wait is long	The app stays at Team scope. Escalate to the enablement lead; the target is 5 working days (A-042). An exception can change when or how a gate is met, never skip it

If your app builder can't do this

A builder that can't keep an app private isn't approved at any tier ([Floor 2](#); enforced through [approving tools](#)). Move the app or retire it. For other gaps, such as no separate test and production, no backup export or no audit log, work in this order:

1. Use a compensating control, such as a scheduled export to company storage, or a manual change log. It can cover a good-practice line. For a gate it can only change *how* the gate is met, through a [bounded exception](#); it never replaces the gate.
2. Move to another approved hosting pattern. [Hosting](#) lists patterns that need no build pipeline, such as a low-code tenant or an internal site behind company sign-in.
3. Lower what the app does or holds, then re-answer the [six questions](#); minimization lowers the tier. If the answer stays but controls offset it, use the [downgrade path](#).
4. Don't go live at that tier. Keep the app at a tier whose gates it meets, or retire it.

No pipeline is assumed anywhere on this page.

Accessibility

Accessibility is a legal duty in some situations and good practice in the rest. The technical target is WCAG 2.2 level AA. As of October 2026, no law this page found names AI-built apps; what matters is who uses the app and who you are.

- **Gate, Important and Critical:** an accessibility check (WCAG 2.2 AA) and an AI-disclosure check if people outside the business use the app (Q2(c)) or receive messages it sends or drafts ([verify gates](#)).
- *Good practice*, Important and Critical apps only staff use: WCAG 2.2 AA checks on the one to three core tasks the app exists for, end to end.
- *Good practice*, Team: keyboard use, form labels and contrast.

The law can require more than the gate, such as an accommodation for an employee who must use an internal app (ADA Title I). Record what you tested, when and who did it. Don't publish a blanket claim that the app is legally accessible; tests show what you checked, not what a court would decide.

Which laws may apply?

Situation	Law or rule	What it says	Standard
Customers or the public use the app, and your business is open to the public (US)	ADA Title III	DOJ says websites must be accessible but sets no standard by regulation. ⁵ The Ninth Circuit applied Title III to a website and app tied to physical restaurants, leaving sites with no such link open; circuits differ. ⁶ California's Unruh Act adds minimum damages of \$4,000 per offense ⁷	WCAG 2.2 AA (good practice)
Employees must use the app (US, 15 or more employees)	ADA Title I	Reasonable accommodation, and no screening-out selection criteria; no WCAG mandate for internal tools ⁸	WCAG 2.2 AA (good practice), plus an accommodation route
A state or local government (C2)	ADA Title II web rule	WCAG 2.1 AA from 26 Apr 2027 (population 50,000 or more) or 26 Apr 2028 (smaller entities) ⁹	WCAG 2.1 AA
A US federal agency, or a contractor whose contract requires it	Section 508	The Revised 508 Standards (WCAG 2.0 AA) ¹⁰	WCAG 2.0 AA
A recipient of US HHS funding (for example a hospital or clinic)	Section 504 (45 C.F.R. Part 84)	WCAG 2.1 AA from 11 May 2027 (15 or more employees) or 10 May 2028; other Section 504 duties apply now. Reach to employee-facing content is for counsel ¹¹	WCAG 2.1 AA
A Canadian federal public-sector body, of any size	Accessible Canada Act and SOR/2025-255	New or updated web pages, including employee-facing ones, from 5 Dec 2027 ¹²	CAN/ASC-EN 301 549:2024
A federally regulated private-sector business in Canada with 100 or more employees (99 or fewer are exempt)	Accessible Canada Act and SOR/2025-255	Public and employee-facing web pages from 5 Dec 2028, plus training for staff who build or buy digital technology; employee "collaborative digital environments" are exempt (whether a worker-built app counts is for counsel) ¹²	CAN/ASC-EN 301 549:2024
An Ontario business or employer	AODA (O. Reg. 191/11)	With 50 or more employees, public websites meet WCAG 2.0 AA except live captions and audio description; internal sites aren't covered. Every Ontario employer, including those with fewer than 50 employees (since 1 Jan 2017), provides job information in an accessible format on an employee's request (s. 26) ¹³¹⁴	WCAG 2.0 AA (public); accessible formats on request
You sell covered consumer services into the EU	European Accessibility Act (Directive 2019/882)	Applies since 28 June 2025, with transitional rules to 2030; microenterprises providing services are exempt; transposition varies ¹⁵	EN 301 549 / WCAG

WCAG 2.2 AA generally covers 2.1 and 2.0 AA, because W3C designed 2.2 to be backward compatible,¹⁶ so one target works across the table.

What AI app builders commonly get wrong

No study we found scans named AI app builders' output for WCAG 2.2, so we give no rate for any builder; on the web generally, 95.9% of a million home pages had detectable failures, mostly contrast, alt text, form labels, empty links and buttons, and page language.¹⁸ Model-generated interfaces also fail in ways automated checkers pass, such as vague labels and headings that don't match content,¹⁹²⁰ so run an automated scan, then a manual keyboard and screen-reader pass using the [ten accessibility checks \(CSV\)](#), which include the four new WCAG 2.2 AA criteria.¹⁷

Don't: publish an accessibility claim nobody tested

A statement or badge that says the app is accessible, with no test behind it, is a risk on its own. Record the checks you ran.

Go-live sign-off

Each line is ticked by the person in its "Who checks" column. The app owner then confirms the checklist, and from Important the technical reviewer signs the technical check. The registry `status` moves to `active`. Record it once per app with the [decision record template](#).

Illustrative composite

Title: Take the leave-request tracker live · **Status:** accepted · **Date:** YYYY-MM-DD · **Owner:** the HR operations lead (app owner) · **Reviewers:** a technical reviewer; the enablement lead (exception approver)

Context. A tracker holding leave reasons moves to company-wide use. Tier: Important, from Q1(c). Only staff use it, so the accessibility and AI-disclosure gate doesn't apply.

Decision. Go live with every Important gate met except one, under a bounded exception the enablement lead approved: the runbook's restore and rollback sections are due in 30 days, because the named backup is running the restore test first and version history covers rollback meanwhile. The data owner signed off first; that can't be excepted. A deferred screen-reader pass is a good-practice line, so it gets a dated note, not an exception.

Alternatives considered. Delay go-live (rejected: rollback is covered). Go live before the technical check (not allowed).

Review by: the exception's expiry, 30 days from the date above.

Questions to ask your counsel

1. Which accessibility laws apply to the apps our staff and customers must use?
2. Does any contract require Section 508 or WCAG conformance, and for which apps?

3. Does ADA Title III, or a state law such as California's Unruh Act, put our customer-facing apps at risk, including apps with no link to a physical location?
4. What accommodation process do we have for employees with disabilities who must use a worker-built app, and where does the app record name it?
5. Does the HHS Section 504 web rule reach our employee-facing apps, and which other Section 504 duties apply now?
6. Are we federally regulated under the Accessible Canada Act, from which date, and does the "collaborative digital environment" exemption cover any of our apps?
7. How do we meet Ontario's duty to give employees job information in an accessible format on request?
8. If we sell to consumers in the EU, does the European Accessibility Act apply, or are we a microenterprise?
9. What should we say, and not say, about an app's accessibility or use of AI on a public page or in a contract?

Assumptions and evidence

- **A-015** (Medium): we expect most exposures of worker-built apps to be configuration errors that the checklist catches; test this by tracking what the pre-launch exposure scan finds.
- **A-014** (High): we expect AI-builder apps without platform defaults to expose data; test this by counting negative-test failures at go-live.
- **A-042** (Low): we expect technical-check capacity to keep up with Important apps; test this by measuring the median wait against five working days.

Evidence: documented exposure incidents (vendor and news research), scanner documentation, government rules, statutes and one court opinion, one peer-reviewed accessibility study, and a large automated web scan. There is no controlled study of go-live checklists for worker-built apps. Facts on this page are as of October 2026.

Related

- [Check an AI-built app before release](#): the V.E.R.I.F.Y. self-check and the reviews behind the checklist.
- [Security checklist for AI-built apps](#): how to check the access, data, secrets and input lines.
- [Stop vibe-coded apps exposing data](#): the negative test and the no-developer tests.
- [Where to host worker-built apps](#): approved hosting for each app.
- [Document your app](#): the app record and runbook.
- [Support model for worker-built apps](#): who fixes what after go-live.

-
1. UpGuard, research on Power Apps portals exposing about 38 million records to anonymous users; the vendor first called it "by design", 23 Aug 2021. <https://upguard.com/breaches/power-apps> [vendor], accessed 2026-10-21.
 2. NIST NVD, CVE-2025-48757 (insufficient row-level security in Lovable-generated projects; the entry notes the vendor disputes it and says customers are responsible), published 30 May 2025. <https://nvd.nist.gov/vuln/detail/CVE-2025-48757> [primary], accessed 2026-10-21.
 3. OWASP ZAP, "ZAP - Baseline Scan" (one-minute spider by default, then passive scanning only). <https://www.zaproxy.org/docs/docker/baseline-scan/> [primary], accessed 2026-10-21.

4. Escape.tech, methodology post on how it found vulnerabilities in apps built with vibe-coding platforms, 29 Oct 2025 (anonymous keys found in front-end bundles, then the backend APIs queried). <https://escape.tech/blog/methodology-how-we-discovered-vulnerabilities-apps-built-with-vibe-coding/> [vendor], accessed 2026-10-21.
5. US Department of Justice, "Guidance on Web Accessibility and the ADA", 18 Mar 2022 (no regulation sets a web standard for Title III). <https://www.ada.gov/resources/web-guidance/> [regulator], accessed 2026-10-21.
6. Robles v. Domino's Pizza, LLC, 913 F.3d 898, No. 17-55504 (9th Cir. Jan. 15, 2019), certiorari denied Oct 2019; opinion read in full (nexus to physical restaurants "critical to our analysis"; reversed a dismissal and remanded without deciding whether the site violated the ADA; fn. 6 leaves open sites with no nexus to a physical place). <https://cdn.ca9.uscourts.gov/datastore/opinions/2019/01/15/17-55504.pdf> [court], accessed 2026-10-21.
7. California Civil Code §52(a) (minimum damages of \$4,000 per offense) and §51(f) (an ADA violation is an Unruh Act violation). https://leginfo.ca.gov/faces/codes_displaySection.xhtml?lawCode=CIV§ionNum=52 [statute; statute text not re-read in this review], accessed 2026-10-21.
8. 42 U.S.C. §§12111–12112 (ADA Title I: reasonable accommodation unless an undue hardship; selection criteria that screen out people with disabilities only if job-related). <https://www.law.cornell.edu/uscode/text/42/12112> [statute], accessed 2026-10-21.
9. US Department of Justice, interim final rule extending the compliance dates of the ADA Title II web and mobile app rule (smaller entities and special districts from 26 Apr 2028; technical standard unchanged), Federal Register Doc. 2026-07663, 20 Apr 2026. <https://www.federalregister.gov/d/2026-07663> [regulator], accessed 2026-10-21.
10. US Access Board, "Information and Communication Technology (ICT)" (Revised 508 Standards, 36 C.F.R. Part 1194, incorporating WCAG 2.0 AA; meeting 2.2 AA generally also meets it). <https://www.access-board.gov/ict/> [regulator], accessed 2026-10-21.
11. HHS, Section 504 rule extension of compliance dates (fewer than 15 employees from 10 May 2028), Federal Register Doc. 2026-09266, 11 May 2026 (the extension doesn't cover other Section 504 duties, such as reasonable modifications; other agencies' 504 rules differ). <https://www.federalregister.gov/d/2026-09266> [regulator], accessed 2026-10-21.
12. Government of Canada, Regulations Amending the Accessible Canada Regulations, SOR/2025-255, registered 5 Dec 2025, Canada Gazette Part II, 17 Dec 2025: ss. 19.1 and 19.4(4) (collaborative digital environments), ss. 19.4–19.5 (web pages), s. 18 (coming into force), and the Regulatory Impact Analysis Statement (federal public-sector web pages from 24 months; all other requirements from 36 months); Accessible Canada Act, S.C. 2019, c. 10. Covers banks, telecoms, airlines and interprovincial transport. <https://gazette.gc.ca/rp-pr/p2/2025/2025-12-17/html/sor-dors255-eng.html> [statute], accessed 2026-10-21.
13. Government of Ontario, "How to make websites accessible" (AODA, O. Reg. 191/11; the website duty excludes success criteria 1.2.4 and 1.2.5). <https://www.ontario.ca/page/how-make-websites-accessible> [regulator], accessed 2026-10-21.
14. University of Waterloo Library, AODA toolkit (employment standards, including accessible formats for employees, apply to organizations with 1–49 Ontario employees from 1 Jan 2017). <https://uwaterloo.ca/library/aoda-toolkit/node/15> [secondary; s. 26 not re-read on e-Laws, which didn't render], accessed 2026-10-21.
15. European Union, Directive (EU) 2019/882 on the accessibility requirements for products and services, Arts. 3(23) and 4(5) (microenterprises: fewer than 10 persons and turnover or balance sheet of EUR 2 million or less), 31(2) (applies from 28 June 2025) and 32 (transitional rules to 28 June 2030); read on an official copy of the EU text, as the EUR-Lex page didn't render. <https://eur-lex.europa.eu/eli/dir/2019/882/oj> [statute], accessed 2026-10-21.
16. W3C, "WCAG 2.2" (backward compatible; removes the obsolete 4.1.1 Parsing criterion). <https://www.w3.org/TR/WCAG22/> [primary], accessed 2026-10-21.
17. W3C, "What's New in WCAG 2.2" (new AA criteria: Focus Not Obscured (Minimum), Dragging Movements, Target Size (Minimum), Accessible Authentication (Minimum)). <https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/> [primary], accessed 2026-10-21.
18. WebAIM, "The WebAIM Million" (2026 report, published Mar 2026 from a Feb 2026 crawl: six error types make up 96% of errors). <https://webaim.org/projects/million/> [research], accessed 2026-10-21.

19. Calò, Gurita and De Russis, "Measuring the Semantic Accessibility Gap in LLM-Generated Web UIs", CHI '26, Apr 2026 (541 semantic violations in 300 interfaces from three commercial models). <https://iris.polito.it/handle/11583/3008330> [research], accessed 2026-10-21.
20. Yoon et al., "A11yn" (arXiv preprint; base-model output often breaks WCAG while looking visually coherent), submitted Oct 2025. <https://arxiv.org/abs/2510.13914> [research], accessed 2026-10-21.

Prompt to Prototype, Make a vibe-coded app production ready, <https://prompt-to-prototype.com/lifecycle/production-ready/>, as of 2026-10-22. Licensed [CC BY-SA 4.0](#).