



Corcava

TEMPLATES • CLIENT OPS TOOLKIT

Client Portal Requirements Checklist

A practical checklist to choose or build the right client portal - permissions, approvals, visibility, file handling, and reporting.

[Download Scoring Sheet](#)

[Download PDF Copy](#)

What's inside

- **Requirements checklist** across access control, approvals, files, visibility, billing, messaging, audit trail, and reporting.
- **Scoring worksheet** to compare tools (minimum viable vs ideal) with weights and notes.
- A quick way to spot **deal-breakers** before you onboard clients into a portal.
- Guidance on how to roll this out without slowing delivery.

Last updated: 2026-01-10

Client portal requirements checklist

Use this before you buy or build a client portal. The goal is simple: clients can find the latest work, approve quickly, and always know what happens next.

Quick guidance

Start with a minimum viable portal that supports clean access control, clear approvals, and a single source of truth for files. Then add automation and reporting only where it reduces client friction or saves your team time.

Minimum viable vs ideal (at a glance)

Category	Minimum viable portal	Ideal portal
Access control	Roles + permissions. Client sees only their projects and files.	Granular permissions (project, folder, item). Guest access + expiring links.
Files & versioning	Single source of truth for latest files + clear naming.	Version history, approvals tied to file versions, and structured handoffs.
Approvals	One-click approve / request changes with comments.	Approval workflows (stages), assignees, deadlines, and audit trail.
Status visibility	Simple status board: planned, in progress, blocked, done.	Milestones, dependencies, SLAs, and automated client updates.
Billing	Link invoices externally and note payment status.	Billing inside the portal: invoices, retainers, payment reminders.
Messaging	Central place for decisions and questions.	Threaded conversations by item, @mentions, and notification controls.
Audit trail	Who approved what and when (at least per milestone).	Full activity log: comments, uploads, approvals, edits, and exports.
Reporting	Basic weekly update and delivery summary.	Dashboards by client/project + exportable reports and time-based views.

Requirements checklist (core)

Treat these as your non-negotiables. If a portal fails here, it will create client confusion and internal rework.

Access control

- Clients can only see their own work (projects, tasks, files, and updates).
- Clear roles (admin, team member, client) and a predictable permission model.
- Guests/external reviewers can be invited safely when needed.
- Notifications are configurable (avoid spamming clients on every minor change).
- Offboarding is simple (revoke access without breaking your internal archive).

Files & versioning

- One place to find the latest files (no “final_v7” scavenger hunts).
- Version history or explicit file versioning for approvals and handoffs.
- Consistent folder structure and naming conventions you can reuse per client.
- Supports links/embeds to tools your team uses (Drive, Figma, etc.).
- Fast client access (downloads work, permissions don't block basic viewing).

Approvals

- Approve / request changes with a timestamp and a clear status per deliverable.
- Comments attach to a specific version (so feedback does not drift).
- You can assign an approver and set due dates (even if reminders are manual).
- Approval history is visible to the team (avoid “who approved this?” debates).
- You can prevent unapproved work from being treated as final.

Status visibility

- Client can see current status and next step without asking you.
- Milestones and target dates are visible and easy to update.
- Blockers and decisions are documented where both sides can find them.
- A weekly update can be generated quickly (manual is fine; consistent matters).
- Clients can see what is waiting on them (what needs review/approval).

Requirements checklist (operations)

These requirements protect margin and reduce churn. Prioritize what matches your business model (retainer, project, or ongoing support).

Billing & retainers

- Invoices/links can be stored per client and tied to projects or milestones.
- Retainer usage or “hours used” can be tracked somewhere clients can understand.
- Purchase order / billing contact info is easy to find.
- Billing status is visible internally (paid, overdue, on hold).

Messaging & decisions

- Messages and decisions live with the work (deliverable/task) - not scattered across email.
- Threads or item-level comments keep context intact.
- @mentions or assignments can route questions to the right owner.
- Client notifications are controlled (less noise, more signal).

Audit trail

- You can answer: who uploaded, who changed, who approved, and when.
- Activity log is easy to search when a dispute happens.
- Exportable log or history exists for compliance / client records.

Reporting

- Client-friendly weekly summary (progress, next steps, risks).
- Internal view for delivery (capacity, overdue items, upcoming approvals).
- Exports (PDF/CSV) for client stakeholders when needed.

Red flags to avoid

If you cannot clearly separate client access, keep approvals tied to versions, or reliably find the latest files, the portal will create more support work than it removes.

Scoring worksheet (printable)

Give each requirement a score from 0-5, multiply by the weight, and compare totals across tools. Use notes to capture deal-breakers.

How scoring works

Weighted score = Weight × Score. Suggested default weights are included - change them based on your delivery model.

Category	Requirement	Weight	Score	Notes
Access control	Clients can only see their own work	3		
Access control	Granular permissions / guest access	2		
Files	Single source of truth for latest files	3		
Files	Version history tied to approvals	3		
Approvals	Approve / request changes per deliverable	3		
Approvals	Approver assignment + due dates/reminders	2		
Status	Milestones and next steps visible to client	3		
Status	Client sees what is waiting on them	2		
Messaging	Threaded comments by item / deliverable	2		
Audit trail	Searchable history of actions/approvals	2		
Reporting	Weekly summary + exports	2		
Billing	Invoices/retainers visible where needed	1		

Total weighted score: _____

Tip: score three tools side-by-side in the Google Sheet to avoid recency bias.

Next steps

A portal works when it becomes part of your delivery system (kickoff, approvals, handoffs, and weekly updates). Use this page to turn the checklist into a habit.

How to roll this out

- Pick your **minimum viable** requirements (the ones that prevent support chaos).
- Set default weights (1-3) and agree on deal-breakers as a team.
- Pilot the portal with one client and tighten your folder structure + approval flow.
- Add reporting/automation only after the basics feel effortless.

Related templates

- **Figma Handoff Checklist:** corcava.com/templates/client-ops/figma-handoff-checklist
- **Project Kickoff Checklist:** corcava.com/templates/client-ops/project-kickoff-checklist
- **Change Request Form Template:**
corcava.com/templates/client-ops/change-request-form-template
- **Scope of Work Template:** corcava.com/templates/client-ops/scope-of-work-template
- **Creative Brief Template:** corcava.com/templates/client-ops/creative-brief-template

Tip

If you publish this on the site, add a short “minimum viable portal” section near the top of the HTML page and link directly to your scoring sheet download.