A PRACTICAL GUIDE

The Freelance Developer's Handoff Checklist

A practical guide to smooth project transitions and client support

Made by the Braxis machine

braxisai.com

ABOUT THIS GUIDE

What this is

This guide is for solo freelance developers.

It is a practical method, not a promise of a result. It contains no invented statistics, no fake customers, and no guarantees. The words were drafted with AI assistance.

Use the checklist at the end on a real piece of work this week.

A PRACTICAL GUIDE

Documentation: Beyond Code Comments

Start with a high-level overview document outlining the project’s purpose, key features, and technical architecture. Include architecture diagrams—even simple flowcharts—because they clarify dependencies and data flows. For databases, provide schema diagrams with field descriptions and relationships. Document third-party services with API endpoints, authentication methods, and rate limits. Write clear setup instructions, including environment variables, database migrations, and deployment steps. Use a version-controlled folder (like `/docs`) for all handovers. Include a troubleshooting section with common errors and solutions. Avoid vague notes; specify exact commands or paths. If using cloud services, document access permissions and billing alerts. This isn’t just for clients—it’s your future self’s insurance.

A PRACTICAL GUIDE

Access and Permissions: Lock It Down

Before handoff, audit all access credentials and revoke unnecessary permissions. Create a secure, time-limited access log for the client, including database usernames/passwords, API keys, and admin panel credentials. Use a password manager to generate unique, complex passwords for each project. Document where each credential is stored (e.g., environment variables, secrets manager) and how to reset them. For cloud services, ensure the client has only the minimum required permissions—no overprivileged accounts. Provide clear instructions on how to rotate credentials independently. If using third-party tools, explain how to revoke access (e.g., OAuth tokens). Never email credentials directly; use encrypted channels or a secure handoff platform. Label each credential with its purpose and expiration date if applicable.

A PRACTICAL GUIDE

Support Window: Set Clear Expectations

Define the support window upfront in a written agreement or email. Specify the duration (e.g., 30 days) and scope (e.g., critical bugs only). Document what constitutes a support request—limit it to issues directly tied to the delivered work. Provide a contact method (email, ticket system) and response time (e.g., 24 hours). For urgent issues, outline an escalation path if the client's support team can't resolve them. Include a disclaimer about maintenance costs for future changes. If offering post-handoff support, charge a retainer or hourly rate upfront. Avoid vague promises; specify what's included (e.g., bug fixes) and what's excluded (e.g., feature requests). Use a template to standardize this process across clients.

A PRACTICAL GUIDE

Testing and QA: Leave No Stone Unturned

Conduct a final test cycle with the client to ensure all features work as intended. Create a test plan covering core functionality, edge cases, and integration points. Document any known issues and their workarounds. For web apps, test across browsers and devices; for APIs, verify endpoints and payloads. If using automated tests, include instructions on how to run them. For databases, validate data integrity and backup procedures. Test security measures like authentication flows and input validation. Provide the client with a checklist of tests to run post-handoff. If the project involves user accounts, explain how to reset passwords or recover lost data. Leave a note about any pending tasks or open questions—this prevents surprises later. Test everything yourself one last time before handoff.

A PRACTICAL GUIDE

Final Walkthrough: Close the Loop

Schedule a live walkthrough to demonstrate the project’s functionality and answer questions. Walk the client through key features, explain how to use them, and clarify any ambiguities. Record the session (with permission) as a reference for future support. Provide a summary email with key takeaways, including links to documentation, credentials, and support contacts. Confirm in writing that the client has reviewed the handoff materials and understands their responsibilities. Get a signed acknowledgment (even if informal) that the project has been accepted. If the client requests changes post-handoff, document them separately and charge accordingly. End with a clear next step—whether it’s signing off, paying the final invoice, or scheduling maintenance. This step ensures alignment and reduces post-handoff friction.

USE THIS

Checklist

☐ 1. Create a high-level overview document with architecture diagrams, setup instructions, and troubleshooting guides.
☐ 2. Audit and document all credentials, storing them securely and providing clear instructions for access and rotation.
☐ 3. Define a support window in writing, specifying scope, response times, and escalation paths.
☐ 4. Conduct a final test cycle with the client, documenting tests, known issues, and workarounds.
☐ 5. Schedule a live walkthrough, provide a summary email, and obtain written confirmation of project acceptance.

Where this came from

This guide is a digital product from Braxis AI. The machine wrote it with AI assistance. It contains no invented statistics, no fake customers, and no guarantees. The PDF is delivered immediately after checkout. If something is wrong, email braxisai@gmail.com and it will be fixed.

braxisai.com