

CODE WITH PURPOSE

Learn. Build. Give Back.

New Project Pre-Start Checklist

**A Reusable Template — Use Before Starting
Any New Project**

Setup, research, and planning steps to complete before writing the first line of code — for any project, any domain, any client.

How to Use This Template

Use this checklist at the very start of every new project — client work, portfolio builds, automation systems, websites, anything under Code With Purpose. It forces you to define the problem and do the necessary research before touching a keyboard for the build itself. Photocopy this mentally (or literally re-fill this PDF) for every new project going forward.

Project Name: _____

Client / For (self / client name):

Start Date: _____

Target Completion Date: _____

What's Inside

1. Problem Definition Checklist
2. Domain & Market Research Checklist
3. Technical Research Checklist
4. Competitor & Reference Research Checklist
5. Scope, Tools & Environment Setup Checklist
6. Data / Content / Asset Research Checklist
7. Risk, Legal & Constraint Checklist
8. Planning & Timeline Checklist
9. Documentation & Branding Checklist
10. Final Go/No-Go Checklist — Ready to Start Building?

1

Problem Definition Checklist

Every project — technical or not — fails or succeeds based on how clearly the problem was defined before work started. Do not skip this step even when a project feels simple.

- ☐ **Problem statement** — Write a one-paragraph problem statement: what problem does this solve, and for whom?
- ☐ **Success criteria** — Define what success looks like in concrete, measurable terms — not "make it good" but a specific number, outcome, or deliverable.
- ☐ **End user / audience** — Identify who the end user or audience is, and what they actually need (not what you assume they need).
- ☐ **Out-of-scope list** — List what is explicitly OUT of scope, so scope creep doesn't quietly expand the project later.
- ☐ **Deliverable confirmed** — If this is client work, confirm the deliverable format and deadline in writing before starting (email, message, or contract).

Before building anything, understand the world the project lives in — the industry, the audience's expectations, and what "good" looks like in that space.

- ☐ **Domain vocabulary** — Research the industry or domain this project belongs to — key terminology, common practices, and what problems people in this space usually face.
- ☐ **Existing examples** — Identify 2–3 real examples of similar projects, products, or businesses already doing something close to this — note what they do well and what's missing.
- ☐ **Audience research** — If there's a target audience or customer, research their actual behavior and expectations (forums, reviews, social media, direct conversation if possible).
- ☐ **Current trends** — Note any current trends, seasonal factors, or recent changes in this space that could affect the project's relevance.
- ☐ **Pricing / value benchmark** — If relevant, research pricing norms or typical costs in this space — for services, products, or comparable freelance work.

Know your stack and its limitations before committing to it. Switching tools mid-project is far more expensive than a day of research upfront.

- ☐ **Tech stack list** — List every tool, language, framework, or platform this project will require, and confirm you either know them or have budgeted time to learn them.
- ☐ **Read official docs first** — For any new tool or library, read its official "getting started" documentation before writing project code with it.
- ☐ **Known limitations check** — Check for known limitations, deprecations, or common gotchas in your chosen stack by searching "[tool name] common problems" or checking recent GitHub issues.
- ☐ **Hardware / hosting check** — Confirm your machine/environment can actually handle the project's compute, storage, or hosting requirements.
- ☐ **Version control plan** — Decide your version control and backup strategy before writing the first file — not after something is lost.

- ☐ **3 competitors identified** — Find 3 direct or indirect competitors/alternatives and study what they do well.
- ☐ **Gap analysis** — Note specific gaps or weaknesses in those alternatives that this project could do better.
- ☐ **Reference/inspiration folder** — Save 3–5 reference examples (screenshots, links, repos) that represent the quality bar you're aiming for.
- ☐ **Hands-on testing** — If applicable, test the competitor's product/service yourself firsthand rather than only reading about it.

- ☐ **Project folder created** — Set up a dedicated project folder / repository with a clean structure before adding any files.
- ☐ **Tools installed & verified** — Install and verify every required tool, library, or account access works before relying on it mid-build.
- ☐ **README started** — Create a README (even a rough one) stating the project's purpose — you'll flesh it out later, but starting it now keeps scope honest.
- ☐ **Task tracking system chosen** — Set up whatever tracking system you'll use to manage tasks for this project (Notion, Trello, a simple checklist doc — anything, as long as it's decided).
- ☐ **Client communication plan** — If working with a client, confirm communication channel and expected update frequency (e.g. WhatsApp updates every 2 days).

Applies whether the project needs a dataset, written content, images, branding assets, or client-supplied material.

- ☐ **Source inventory** — Identify every data source, content source, or asset this project depends on, and confirm you can actually access it.
- ☐ **License / rights check** — Check licensing or usage rights on any external data, images, fonts, or content before using it in a deliverable — especially for client or public work.
- ☐ **Data quality check** — For any dataset: check its size, quality, completeness, and whether it actually answers the problem defined in Section 1.
- ☐ **Raw backup created** — Back up all raw source material in its original form before you start editing or transforming it.

- ☐ **Biggest risk identified** — Identify the biggest risk that could derail this project (technical, timeline, client, or personal bandwidth) and note a fallback plan.
- ☐ **Legal / copyright check** — Confirm there are no legal, licensing, or copyright issues with anything you plan to use or publish.
- ☐ **Data privacy check** — If handling personal or sensitive data (client info, user data), confirm how it will be stored and protected.
- ☐ **Time budget check** — Set a realistic time budget for this project against your other current commitments (job applications, other clients, teaching cohort, etc.) so it doesn't silently stall.

- ☐ **Milestones drafted** — Break the project into major milestones with rough dates for each.
- ☐ **Hardest part identified** — Identify the single hardest or riskiest part of the project and tackle it early — not last.
- ☐ **Time blocked on calendar** — Block real calendar time for this project rather than relying on "whenever I get free time."
- ☐ **Definition of done set** — Decide upfront what "done" looks like, so you don't keep polishing indefinitely.

- ☐ **Documentation method chosen** — Decide how this project will be documented as it's built (commit messages, a build log, or a running notes file).
- ☐ **Brand consistency check** — If this is a portfolio or brand project, confirm it follows Code With Purpose visual identity (navy and gold, "Learn. Build. Give Back.") where relevant.
- ☐ **Showcase plan** — Plan how you'll showcase this project afterward — website, LinkedIn, resume, YouTube — before you're scrambling to write it up after the fact.

If every box below is honestly checked, you're ready to start building.

- ☐ I can explain this project's purpose and success criteria in two sentences.
- ☐ I understand the domain/industry well enough to speak about it confidently.
- ☐ I know my full tech stack and have verified the tools actually work.
- ☐ I've reviewed at least 3 competitors or references for this type of project.
- ☐ My project folder, repo, and tracking system are already set up.
- ☐ I've checked licensing/legal issues on every data source or asset I'll use.
- ☐ I have a rough timeline with milestones and know the hardest part to tackle first.
- ☐ I know exactly how I'll document and showcase this project once it's done.

"Learn. Build. Give Back." — Code With Purpose