



**Gearbly**  
Precision Tech. Human Ability.

## THE GEARBLY AI IMPLEMENTATION MATRIX

**Version:** 2.0 | **Focus:** Enterprise & Agency Workflows

**Objective:** Maximize velocity without introducing architectural debt.

### 1. The Coding Core (IDE & Logic)

*The engine room. Tools used directly within the code editor.*

Task	Primary Tool (Standard)	Pro Alternative (Advanced)	Gearbly Recommendation
General Auto-Complete	GitHub Copilot	Supermaven	Use <b>Copilot</b> for general consistency. It integrates best with VS Code and has the largest training set.
Refactoring Legacy Code	Cursor (IDE)	Sourcery	Use <b>Cursor</b> . Its ability to index your local codebase reduces hallucinations when refactoring complex proprietary logic.
SQL Generation	DataGrip AI	Text-to-SQL (ChatGPT)	<b>DataGrip</b> is safer. It sees your actual schema. Never paste raw SQL from ChatGPT without verifying table names.

### 2. Frontend & UI Engineering

*Accelerating the "Pixel to Code" pipeline.*

Task	Primary Tool	Best Use Case	Risk Level
Component Generation	v0.dev (by Vercel)	Rapidly generating accessible React/Tailwind components (e.g., "Build a pricing table with a toggle").	● Low (Code is clean)
Design to Code	Locofy.ai	Converting Figma files directly into React code. Good for static pages.	○ Medium (Check responsiveness)



<b>Visual Testing</b>	<b>Percy / Chromatic</b>	AI comparison of UI screenshots to detect "visual bugs" after updates.	⦿ Low (Essential for Agencies)
-----------------------	--------------------------	--	--------------------------------

### 3. Backend & Automation

*Handling data flow and server logic.*

Task	Primary Tool	Best Use Case	Gearbly Note
<b>Workflow Automation</b>	<b>n8n</b>	connecting APIs (e.g., Stripe → Google Sheets → Slack) without writing boilerplate script.	<b>Self-Host this.</b> Do not use the cloud version for sensitive client data.
<b>Unit Test Writing</b>	<b>CodiumAI</b>	Auto-generating test cases for Python/JS functions to ensure high coverage.	Mandatory for all "Logic" code.
<b>API Documentation</b>	<b>Mintlify</b>	Scanning code to auto-generate beautiful documentation sites.	Saves hours of manual writing.

### 4. DevOps & Infrastructure

*Security and deployment pipelines.*

Task	Tool	Why we use it
<b>Infrastructure as Code</b>	<b>Terraform (w/ Copilot)</b>	Copilot is surprisingly good at writing standard Terraform modules for AWS/DigitalOcean.
<b>Security Scanning</b>	<b>Snyk</b>	AI-driven vulnerability scanning. It finds dependencies with known hacks before you deploy.
<b>Shell Scripting</b>	<b>Warp Terminal</b>	An AI-enhanced terminal that fixes your bash commands (e.g., "undo that last git commit").



**Gearbly**  
Precision Tech. Human Ability.

## **⚠ The "Do Not Use" List (High Risk)**

*Tools we currently avoid for Client Work due to data privacy or reliability issues.*

1. **Public ChatGPT (Free Tier):** Never paste client API keys, passwords, or PII (Personally Identifiable Information) here. It is used for training.
2. **Fully Autonomous Agents (e.g., AutoGPT):** Good for experiments, but they often get stuck in loops and burn API credits. Do not leave them running unattended on a production server.

## **How to Apply This Matrix**

1. **Junior Devs:** Restricted to **GitHub Copilot** (Autocomplete only) and **v0.dev** (UI).
2. **Senior Engineers:** Authorized to use **Cursor** and **n8n** for architectural workflows.
3. **Review Process:** All AI-generated code must be tagged with a comment `// Generated by AI, Reviewed by [Name]` before merging.