Generative AI Tools Cheat Sheet

Quick Reference for Developers & Teams

1. Introduction

Generative AI is rapidly reshaping the landscape of software development. From creating code snippets on demand to generating realistic images and natural-sounding dialogue, these tools offer new possibilities for developers and teams seeking to accelerate innovation and streamline workflows. Understanding how and when to leverage generative AI can make the difference between keeping pace and leading the way in today's fast-evolving tech environment.

Generative AI refers to systems like language models, image generators, or code engines that can produce new content based on patterns they have learned from massive datasets. Consider tools like ChatGPT, Copilot, and DALL-E: these solutions help automate tasks, spark creativity, and solve complex problems more efficiently than ever.

· **Example:** A developer can use GitHub Copilot to autocomplete code, saving hours on boilerplate and suggesting best practices instantly.

▪ **Example:** A design team might generate dozens of logo concepts using an AI image generator, exploring creative directions before involving human designers.

1.1 Why Generative AI Matters in Software Development

- **Accelerates productivity:** AI can generate code, documentation, and tests, freeing developers to focus on higher-level architecture and problem-solving.

- **Enhances creativity:** By suggesting solutions outside a developer's typical patterns, AI can inspire novel approaches to design and coding challenges.
- **Reduces repetitive tasks:** Routine code generation, refactoring, or even comment writing becomes automated, improving consistency and saving time.
- **Improves collaboration:** Teams can use AI-powered tools to brainstorm ideas, review code, or even translate documentation, bridging gaps between diverse skill sets.

Example: When maintaining a large legacy codebase, AI can help identify potential bugs, suggest refactoring, and automatically generate unit tests, reducing human error and accelerating delivery.

1.2 Purpose of the Cheat Sheet

This cheat sheet is designed as a practical, quick-reference guide for developers and teams who want to:

- Identify the right generative AI tools for specific development tasks (e.g., coding, design, documentation).
- Understand core features, strengths, and limitations of popular tools.
- Adopt best practices for integrating AI into day-to-day workflows.
- Access practical tips, important shortcuts, and usage examples at a glance.

Example: Unsure which AI code assistant is best for JavaScript versus Python? Flip to the relevant page for a side-by-side comparison and recommendations.

1.3 How to Get the Most Out of This Cheat Sheet

- **Keep it handy:** Print it out or bookmark the digital version for quick access during development sprints or brainstorming sessions.
- **Update regularly:** Generative AI evolves quickly, make notes or add links to newly released tools and features as you discover them.
- **Collaborate:** Share insights and tips with your team, and encourage others to contribute their experiences with different tools.
- **Experiment:** Don't hesitate to try out new features or AI-powered tools in side-projects or hackathons to discover their potential firsthand.
- **Evaluate results critically:** Use the cheat sheet as a starting point, but always validate AI-generated content and assess its suitability for your use case.

Example: If an AI tool generates code that compiles but doesn't fit your architectural standards, use the cheat sheet's troubleshooting tips to adjust prompts or refine tool settings.

By combining concise explanations, real-world examples, and actionable tips, this cheat sheet empowers developers and teams to harness the power of generative AI with confidence and creativity.

2. Tool Comparison Table

Tool Name	Best For	Supported Languages	Integration Options	Pricing	Unique Features
GitHub Copilot	Rapid code completion and suggestions within common IDEs	JavaScript, Python, TypeScript, Go, Ruby, and more	Visual Studio Code, JetBrains IDEs, Neovim	Subscription-based, free for verified students & open-source maintainers	Context-aware code suggestions, autocomplete for multiple languages, natural language to code prompts
Tabnine	AI-powered autocomplete for	Python, JavaScript, Java, C++,	VS Code, JetBrains, Sublime, Atom,	Free tier, Pro plan with advanced features	Private model training, team-based

	diverse development environments	PHP, and more	Eclipse, Vim, Emacs		knowledge sharing, privacy-focused options
Amazon CodeWhisperer	Developers working with AWS services and cloud-based projects	Python, Java, JavaScript, TypeScript, C#, Go, Ruby, and more	VS Code, JetBrains, AWS Cloud9	Free individual tier, paid for professional/enterprise	Security scanning for vulnerabilities, AWS service integration, reference tracking for code suggestions
Google Gemini Code Assist	Developers seeking Google ecosystem integration	Python, Java, JavaScript, Go, Dart, and more	Cloud Shell, Google Colab, VS Code	Currently in preview/beta, pricing details forthcoming	Seamless Google integration, code explanation, and

	and code review				documentation generation
Replit Ghostwriter	Collaborative and browser-based coding	Python, JavaScript, C, C++, Java, HTML/CSS, and more	Replit online IDE	Subscription with free trial available	In-browser AI assistant, collaborative coding, code debugging and explanations
Kite	ML-powered code completions for data science and scripting	Python (primary), limited support for Go, Java, JavaScript, C, C++	VS Code, PyCharm, Atom, Sublime, Vim, IntelliJ	Discontinued as of 2022 (listed for reference)	Deep learning-based suggestions, documentation lookups, focused on productivity

					for Python users

3. Key Tool Highlights

- **GitHub Copilot** – An AI pair programmer integrated tightly with major editors like VS Code. Copilot leverages OpenAI models to provide context-aware code completions, real-time suggestions, and even entire function or file-level scaffolding based on natural language comments. Its deep integration enables faster prototyping and improved developer productivity.
- **Amazon CodeWhisperer** – Designed for AWS and cloud-centric workflows, CodeWhisperer offers real-time code generation with built-in security scanning. It provides relevant code recommendations for AWS APIs, infrastructure-as-code templates, and security best practices, making it a strong companion for cloud developers aiming to build and deploy safely.
- **Tabnine** – Focused on privacy and team customization, Tabnine uses AI to generate code completions, with the option to train custom team models for consistent style and confidentiality. It works with most popular IDEs and supports

a wide range of programming languages, catering to both individual and enterprise-level coding needs.

- **CodiumAI** – Specializing in automatic unit test creation, CodiumAI helps developers write robust, reliable code by suggesting and generating intelligent test cases. Its AI-driven analysis of code logic ensures greater test coverage and helps catch edge cases early in the development process.
- **DeepCode by Snyk** – An advanced tool for static analysis, DeepCode leverages AI to detect bugs, code smells, and security vulnerabilities in real time. Integrated with Snyk's security platform, it continuously scans repositories, surfaces actionable insights, and helps teams remediate issues before they reach production.
- **Mintlify** – Mintlify streamlines code documentation by generating clear, structured, and up-to-date documentation directly from codebases. Its automatic parsing and formatting reduce manual effort, ensuring that documentation is both accurate and developer-friendly, which enhances onboarding and collaboration.
- **ChatGPT** – Serving as a versatile AI assistant, ChatGPT assists developers by offering debugging help, code explanations, and generating pseudocode from natural language queries. Its conversational interface and adaptability make it a valuable resource for learning, prototyping, and troubleshooting in any programming environment.

4. Integration Tips

4.1 Embedding Tools in Your Development Workflow

Effectively embedding AI-powered coding assistants into your workflow involves more than simply installing plugins. Start by identifying which stages of your software development process benefit most from automation—such as prototyping, refactoring, documentation, and testing—and configure relevant tools at those touchpoints. For example, integrating GitHub Copilot or Tabnine directly into your preferred IDE (like VS Code, JetBrains, or Atom) ensures real-time access to code completions and suggestions as you type.

Consider building toolchains: connect documentation generators like Mintlify with your version control system so docs update automatically with each commit, or use CodiumAI to generate tests alongside new features. Leveraging IDE extensions and configuring API keys securely will help ensure these tools are seamlessly embedded, reducing context switching and maintaining developer flow.

4.2 Combining Multiple AI Assistants for Maximum Value

No single AI assistant covers every aspect of the software lifecycle. To maximize value, strategically combine tools based on their strengths. For instance, use CodeWhisperer for cloud and AWS-centric code generation, while relying on DeepCode by Snyk for security scanning and static analysis. Meanwhile, ChatGPT can supplement these tools by answering conceptual questions, providing debugging guidance, or generating pseudocode for complex logic.

Establish a workflow where initial code suggestions come from Copilot or Tabnine, security validation from DeepCode, documentation from Mintlify, and robust test coverage from CodiumAI. This layered approach ensures broad coverage—from ideation and implementation to validation and documentation—without overloading your workspace. Keep integrations modular, enabling you to swap in new tools as needs evolve.

4.3 Maintaining Code Quality While Using AI

While AI assistants can boost productivity and catch many issues, maintaining high code quality still requires human oversight. Regularly review AI-generated code for correctness, readability, and adherence to your team's style guides. Pair AI suggestions with automated code linters and static analysis tools to surface potential bugs early.

Establish code review practices where peers and AI-driven tools work in tandem: let DeepCode or similar tools highlight vulnerabilities, then perform a manual audit to verify logic and context. For sensitive applications, avoid blindly accepting large blocks of generated code test thoroughly using both CodiumAI-generated and hand-written unit tests.

Finally, promote a culture of continuous learning: encourage developers to treat AI-generated output as suggestions rather than solutions, refining and adapting code to meet evolving standards. By thoughtfully integrating, combining, and vetting AI tools, teams can accelerate development while upholding the integrity and maintainability of their codebase.

5. Quick Start Prompts

Jumpstarting your workflow with tailored prompts can drastically increase the effectiveness of AI coding assistants. Well-crafted prompts not only save time but also lead to more accurate and context-aware outputs, streamlining everything from boilerplate generation to in-depth code explanations.

GitHub Copilot: "Generate a REST API endpoint in Python using Flask."

Use Copilot for rapid scaffolding of web services or repetitive backend logic. By specifying your stack and framework requirements like Flask for Python you empower Copilot to generate ready-to-use endpoint code, complete with route decorators and request handling. For example, typing the above prompt can produce a function complete with input validation and basic response formatting, providing a robust starting point for your API.

ChatGPT: "Explain this SQL query in plain English."

Leverage ChatGPT when you encounter complex or legacy SQL statements that need clarification. Paste the query along with this prompt, and ChatGPT will break down its logic, summarize what it retrieves or updates, and translate nested statements into everyday language. This is invaluable for onboarding, code reviews, and ensuring shared team understanding of database operations.

5.1 Additional Prompts for Diverse Scenarios

Tabnine: "Suggest JavaScript code to debounce an input event handler."

Tabnine excels at generating idiomatic snippets quickly. Use concise, specific prompts to receive best-practice implementations for common patterns.

DeepCode by Snyk: "Scan this code for security vulnerabilities."

Run this prompt on critical code paths to surface potential risks, from SQL injection vectors to insecure dependencies.

CodiumAI: "Write unit tests for this new function."

Accelerate test coverage by prompting CodiumAI to generate a suite of relevant unit tests, ensuring new features are validated from the outset.

Mintlify: "Document this class and its public methods."

Quickly generate clean, consistent documentation to keep your codebase maintainable and user-friendly.

5.2 Tips for Crafting Effective Prompts

Be clear and specific about your requirements—mention frameworks, languages, or constraints.

For explanations, provide full code snippets or queries to ensure context-rich answers.

Iterate on prompts if initial outputs miss the mark; a minor tweak can yield vastly improved results.

Combine prompts with in-tool feedback (e.g., Copilot's "accept/modify" suggestions) to refine outputs further.

With a library of practical, targeted prompts at your fingertips, AI coding assistants can shift from occasional helpers to indispensable partners, empowering you to move seamlessly from ideation and implementation to documentation and beyond.

6. Resources & Next Steps

6.1 Link to GSDC Generative AI in Software Development Certification

To deepen your expertise and gain formal recognition, consider pursuing the GSDC Generative AI in Software Development Certification. This certification offers a structured curriculum focused on practical applications of generative AI across the software development lifecycle. Visit the official GSDC site for course details, enrollment, and upcoming session dates. [URL]

6.2 Further Reading & Case Studies

- **Automating Test Generation with AI:** Opportunities and Pitfalls – Explore how leading teams are leveraging generative AI to expand test coverage and accelerate release cycles.

- **AI-Assisted Documentation:** Streamlining Knowledge Sharing – A practical look at how tools like Mintlify and Copilot are transforming software documentation practices.
- **Case Study:** Integrating Generative AI into CI/CD Pipelines – Read about real-world implementations, challenges faced, and measurable outcomes when embedding AI solutions within existing workflows.

For a comprehensive overview, consult research papers and whitepapers from industry conferences such as NeurIPS and ICSE, which frequently feature advancements in AI-driven software engineering.

By leveraging these resources, you can stay ahead of emerging trends and ensure your development practices continue to evolve alongside the rapidly shifting landscape of generative AI.

GSDC
Global Skill Development Council

CERTIFICATION IN GENERATIVE AI IN SOFTWARE DEVELOPMENT

Generative AI Software Development is based on AI-Driven Code Generation, Automation, and Software Innovation.

GSDC
Global Skill Development Council
Generative AI in Software Development
CERTIFIED

ABOUT GSDC CERTIFICATION

LIFETIME VALIDITY

GSDC Certification is an globally accreditted certification with lifetime validity.

EBOOK

Extensive and exclusive Ebook created by world's experts to help you with understanding core concepts.

CREATED BY EXPERTS

GSDC certifications are created and authored by world's leading experts in the field.

LEARNING MATERIALS

Get access to learning materials such as videos, ebooks, templates, and practice exams, which will help you clear the certification exam.

LEARNING OBJECTIVE

- Gain hands-on skills with the generative AI for software development skill certificate to enhance your coding efficiency.
- Boost your career prospects with a globally recognized generative AI for software development professional certificate.
- Master AI tools and techniques through a practical generative AI for software development course.

Enroll now with the code **LEARN20** To avail **20%** discount

Enroll Now

www.gsdcouncil.org

NOVELVISTA
TRANSFORMING SKILLS

ITIL V4 CERTIFICATION | ITIL FOUNDATION TRAINING, COURSE IN INDIA

Ready to Ace Your ITIL v4 Foundation Exam in 2025? NovelVista is a leading provider of ITIL V4 certification, courses and training in India. We provide IT service management training and ITIL certification in India.

PeopleCert

ABOUT CERTIFICATION

ACCREDITED TRAINING PROVIDER

We are an Approved Training Organization (ATO) delivering globally recognized training.

EXPERT-LED TRAINING

Learn from industry-leading experts with real-world experience.

EXAM READINESS SUPPORT

We prepare you thoroughly for official certification exams.

FLEXIBLE LEARNING MODES

Choose from classroom, online, or self-paced training.

LEARNING OBJECTIVE

- **Align service providers with business goals for optimal delivery.**
- **Apply ITIL practices for effective service management.**
- **Adopt the roles and responsibilities of ITIL 4.**
- **Implement the best practices of ITIL 4.**

Enroll In Any Course And Get Up To 40% Discount. Limited-Time Offer

Enroll Now

www.novelvista.com