

AI Developer Tools

— Three New Capabilities

This document explains three new AI-powered tools added to our development platform. Each one automates a routine part of the software development process, saving developer time and reducing the chance of human error.

These tools are used by developers during their normal working day. No special setup is required beyond what is already in place.

February 2026

TOOL 1 — CODE GENERATOR

Command: type /angular-generate in the AI chat

When developers need to create a new building block for the application — for example, a new screen, a data service, or a security check — they have traditionally had to write a significant amount of repetitive setup code by hand. This tool automates that process.

What the developer does

The developer tells the AI three things: what type of building block they need, what they want to call it, and what it should do. That is the entire input required.

Example: “Create a service called *notification* that shows success and error messages across the application.”

What the AI does

- 1. Reads the existing codebase first.** Before writing a single line, the AI looks at two or three similar existing files to understand how the team writes code. This ensures the output matches the team’s coding style exactly, rather than producing a generic result.
- 2. Generates the files.** The AI creates the necessary code files, already following the team’s conventions — the same structure, patterns, and style used everywhere else in the project.
- 3. Validates the output.** The AI automatically checks that what it generated is correct and will not introduce errors before presenting the result to the developer.
- 4. Tells the developer what was created** and what (if anything) they need to do next to connect the new piece to the rest of the application.

Business value

Without the tool	With the tool
Developer spends 15–45 minutes creating boilerplate code, then adjusting it to match the team's patterns	Developer describes what is needed in one sentence; the AI produces ready-to-use code in seconds
New team members may not know all the conventions, leading to inconsistent code	Convention compliance is automatic — new and experienced developers produce the same quality output

TOOL 2 — AI FEATURES IN THE APPLICATION

Powered by Google Gemini — runs on the server

The application itself now has two built-in AI features that developers can use as a reference and extend for real projects. These demonstrate how to safely connect an application to an AI model without exposing any security credentials to the public.

Feature A — Subtask Suggester

A developer or end user enters a task title. The AI reads it and returns three to five clear, actionable steps that would help complete that task. For example, entering “Plan team offsite” might return: book the venue, send calendar invites, arrange catering, prepare the agenda, confirm attendance.

Feature B — Priority Sorter

A list of tasks is sent to the AI. It returns each task labelled as high, medium, or low priority, along with a one-sentence explanation of why. This helps users decide what to work on first when facing a large backlog.

How it works (in plain terms)

The application has a small server component running alongside it. When a user requests an AI feature, the browser asks the server, the server asks Google’s AI (Gemini), and the result comes back. The AI credentials never leave the server — they are never visible in the browser or accessible to the public.

What is needed to activate these features:

An API key from Google (available free of charge at Google AI Studio). The key is set as a private configuration value on the server and never shared. If no key is provided, the application continues to work normally — the AI features simply return no result, with no error shown to the user.

Business value

Capability	Value to the business
Subtask Suggester	Demonstrates how to add AI assistance to any task or workflow in a real product, with a working reference implementation ready to adapt
Priority Sorter	Demonstrates AI-powered decision support — a pattern applicable to any scenario where ranking or categorisation from a list is needed
Server-side security model	Shows the correct, secure way to use AI in a web application — credentials stay on the server, not in the browser

TOOL 3 — AUTOMATED CODE REVIEW

Command: type /ai-code-review in the AI chat

Before a developer finishes their work and submits it for a colleague to review, they can ask the AI to check it first. The AI acts like a senior reviewer who checks the code against a defined set of quality and security standards, then produces a written report.

What the developer does

The developer tells the AI what to review: their latest unsaved changes, a specific file, or the differences between two versions of the codebase. That is all that is needed.

What the AI checks

Category	What is checked
Correctness	Are there any coding errors that would cause the application to fail to build or behave incorrectly?
Code quality	Is the code following defined team standards? Are there unnecessary memory leaks, leftover test logs, or inconsistent patterns?
Security	Are there any API keys or credentials accidentally left in the code? Are there any patterns that could expose data to users who should not see it?
Performance	Are there patterns that could cause the application to update unnecessarily or become slow with larger amounts of data?

What the AI produces

A written report saved automatically to the project, containing:

- **Must Fix** — issues that should be resolved before the work is shared or deployed
- **Should Fix** — code quality issues that are not blocking but should be addressed
- **Suggestions** — optional improvements the developer may choose to act on
- **Positive notes** — things done well, to encourage good practices
- **Overall verdict** — Approved / Approved with comments / Changes requested

Note on scope:

The AI only flags issues in the code that was changed. It does not raise warnings about pre-existing problems elsewhere in the codebase. This keeps the report focused and actionable for the developer.

Business value

Without the tool	With the tool
Junior developers may submit code with avoidable issues, consuming senior reviewer time	Common issues are caught automatically before human review, so senior reviewer time is spent on higher-order decisions
Security checks depend on individual reviewer knowledge and attention	A consistent security checklist is applied automatically to every change, every time
Code quality standards drift over time as team composition changes	Standards are enforced at the point of submission, not retroactively

SUMMARY

Tool	What it does	Who benefits
Code Generator /angular-generate	Produces new code files that match the team's existing patterns, validated automatically	All developers, especially those new to the project
AI Application Features Subtask Suggester & Priority Sorter	Demonstrates and provides a working reference for securely adding AI features to any web application	Product teams exploring AI feature development
Automated Code Review /ai-code-review	Checks code changes for correctness, quality, and security before human review	All developers; frees up senior reviewer capacity