Prompt Templates for Coding Assistants (LLMs)

Overview

LLM-based coding assistants can accelerate development, improve quality, and help troubleshoot and refactor code. But their effectiveness depends on how you prompt them. This guide provides reusable prompt templates and best practices for software engineers building applications across stacks and stages.

# 1. General Guidelines for Effective Prompts

- Be specific: Include language, frameworks, context, and goals.

- Set constraints: Output length, format (JSON, Markdown, Python), or file boundaries.

- Iterate: Start broad, refine with follow-ups.

- Use delimiters: Triple backticks (```) for code, // comments for expectations.

- Request step-by-step reasoning for complex problems.

# 2. Prompt Templates by Use Case

## A. Code Generation

**Template: Basic Feature Request**

You are a senior [language] developer. Write a [function/module/class] to do the following:  
- [Describe what the feature should do]  
- Use [library/framework]  
- Must run on [platform/environment]  
- Include [error handling/logging/etc.]  
Return only the code.

## B. Unit Test Generation

**Template: Generate Unit Tests**

Write unit tests for the following function in [language] using [testing framework].  
Function:  
```[paste code here]```  
Requirements:  
- Cover edge cases  
- Use proper setup/teardown if needed  
- Return only the test file

## C. Debugging & Fixing

**Template: Help Me Debug**

The following [language] code throws an error when [describe issue].  
Code:  
```[code snippet]```  
Error message:  
[paste error]  
Can you help identify and fix the issue? Please explain and show corrected code.

## D. Refactoring

**Template: Refactor for Clarity or Best Practices**

Please refactor the following [language] code for [performance/readability/security/best practices].  
Code:  
```[paste code]```  
Keep the same functionality. If assumptions are made, explain them.

## E. Architecture & Design Assistance

**Template: Suggest Application Structure**

Suggest a project structure for a [type of app] using [language + framework].  
- Features include: [auth/database/API integration/etc.]  
- Must follow [design patterns/style guides]  
- Assume the project will scale  
Provide a folder/file layout and brief explanation.

## F. API Design & Documentation

**Template: Create/OpenAPI Spec**

Generate an OpenAPI 3.0 specification for a REST API with the following endpoints:  
- [GET /items]  
- [POST /items]  
- [GET /items/{id}]  
- [DELETE /items/{id}]  
Assume JSON format. Add basic validation schemas and status codes.

**Template: Generate API Docs from Code**

Based on the following function, generate Markdown-style API documentation.  
Function:  
```[paste code]```  
Include:  
- Summary  
- Parameters  
- Return type  
- Example usage

# 3. Workflow Integration Templates

## G. Code Review & Feedback

**Template: Code Review Comments**

Act as a code reviewer. Review the following pull request diff and provide comments on performance, security, or clarity.  
Diff:  
```diff  
[Paste PR diff or side-by-side code]  
```  
Respond with actionable feedback.

## H. DevOps & Automation Scripts

**Template: Create CI/CD Pipeline**

Write a CI/CD pipeline YAML file for [GitHub Actions / GitLab / Jenkins] to:  
- Run tests  
- Lint code  
- Build and deploy to [platform]  
Language: [Node.js / Python / Java]  
Environment: [Docker / Kubernetes / EC2]

# 4. Bonus: Meta-Prompts to Improve Prompts

**Template: Optimize My Prompt**

Here is my current prompt:  
"[paste prompt]"  
How can I make it more effective or structured to get better code responses from an LLM?

**Template: Self-evaluate Code**

Review this LLM-generated code for issues or improvements.  
Code:  
```[paste code]```  
Highlight:  
- Potential bugs  
- Inefficiencies  
- Better alternatives

# 5. Best Practices Summary

|  |  |
| --- | --- |
| Tip | Why It Helps |
| Include code samples | Context improves accuracy |
| Specify frameworks | Avoids vague/default implementations |
| Set format constraints | Prevents noise in output |
| Ask for explanations | Helps with understanding and learning |
| Iterate | Refine for better accuracy |