

Prompt cheat sheet for GitHub Copilot Enterprise

GitHub Copilot Enterprise enables developers to quickly familiarize themselves with GitHub projects by allowing them to pose natural language questions to their organisation's code repositories. This cheat sheet is designed to offer a variety of sample prompts, helping you reduce the time needed to understand your organisation's code.

To use these prompts, you need to have a valid GitHub Copilot Enterprise license and enter these prompts in GitHub Copilot chat in a browser. (<https://github.com/copilot>) You also need to specify a repo that has been indexed. Remember, not all prompts will be applicable for all repos.

Summarise functionality and code analysis

Quickly understand the main features and overall purpose of the codebase:

Create a summary of the functionality of the code in this repository.

Identify the primary components and their specific roles within the project:

What are the key modules and their responsibilities in this codebase?

Gain a high-level understanding of the application's workflow & architecture:

Can you provide an overview of how this application works?

Identify the technologies and their versions to understand the tech stack:

What technologies are used in this repo and what versions of each are used?

Identify critical areas that may require focus, such as bugs, outdated code, or areas for improvement:

Analyse the code and let me know what in the code needs my attention

Detect parts of the code that might affect the application's performance:

Highlight any potential performance issues in this repo.

Understand how a feature (Eg. Bot message) is implemented:

Where and how is the bot message generation logic implemented?

Gain a general understanding of the code quality by checking if industry best practices are used:

What areas in this code do not adhere to best practices?

Feature suggestions and adding new functionality

Explore potential enhancements and check issues and PRs for existing feature requests:

What features would be good to add to this app? Suggest a feature I should add to this repository? Also, are there any feature requests currently open?

Identify areas where the application can be enhanced:

Based on the current functionality, what improvements can be made?

Incorporate user feedback to improve the application:

Analyse the issues for this repository to see if there are any comments or suggestions of functionality that can be implemented?

Get guidance on implementing new features, including user interaction and email integration:

How would I add functionality that would send a chat to the user's email? It could ask for the user's email via the chat prompt.

Understand the process of adding new payment options to the application:

What steps are needed to integrate a new payment gateway (e.g. Stripe) into this app?

Learn how to implement additional authentication mechanisms:

How can I add a new user authentication method (e.g. Social logins) to this app?

CI/CD

Learn about the build, deployment processes, and environments used:

How is this repo built, run, deployed, and what environments are used?

Understand what is being done in the CI job (execute this prompt on the GitHub Actions results screen):

What types of quality checks are performed in this workflow?

Understand the reason why the CI job failed:

Why is this job failing?

Dependencies

Identify commonly used libraries and frameworks within the project:

Analyse and list out any popular dependencies used by this repository.

Locate the list of dependencies to understand what the project relies on:

Where can I find the dependencies for this project?

Determine the frequency of dependency updates to assess maintenance practices:

How often are the dependencies updated in this project?

Ensure that critical dependencies are current to avoid potential issues:

Is the @types/react dependency up to date?

Verify the version of a specific dependency to ensure it is up to date:

Where can I find the latest version of the pypck dependency – to check if it is up to date?

Get suggestions on important questions to ask about the project's dependencies:

What questions should I ask you about the dependencies in this repository?



Vulnerabilities and Bugs

Identify existing bugs or issues that need attention:

Are there any bugs or known issues in this code I should be aware of?

Focus on resolving security-related issues:

List all the open issues related to security vulnerabilities.

Prioritise fixing the most impactful bugs:

What are the most critical bugs currently open in this project?

Discussions, Issues, Pull Requests, and Commits

Quickly understand the key points of a discussion thread:

Summarise this discussion.

Get a concise overview of any issues within the repository:

Summarise this issue.

Understand the progress and any blockers related to an issue:

What is the current status of this issue?

Review potential fixes or improvements for an issue:

What are the proposed solutions for this issue?

Quickly understand the modifications and enhancements made in a pull request:

Summarise the changes introduced in this PR.

Identify the main benefits and changes brought by a PR:

What are the key improvements in this pull request?

Review the specific parts of the codebase impacted by a PR:

List the files and functions affected by this pull request.

Get a detailed view of the changes made in a particular commit:

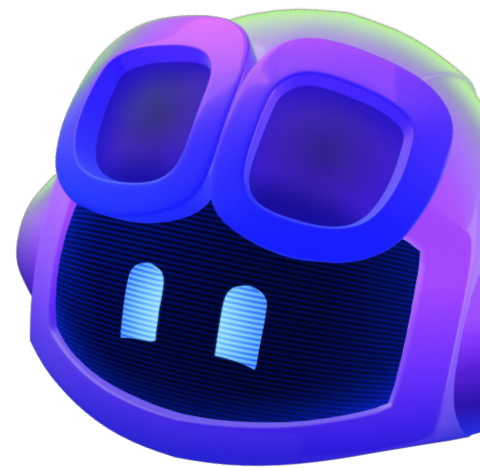
What are the key features and fixes in the latest commit?

Understand the recent updates and fixes in the project:

Summarise the changes in the last five commits.

Identify and review the bugs addressed in recent commits:

What are the main bug fixes in the recent commits?



Generate a First Draft of Business Requirements

Define the scope and details of new features:

Write a detailed business requirement to implement the 'Enhanced Energy Management' feature.

Outline the requirements for implementing multi-language functionality:

Draft a business requirement document for adding multi-language support.

Create a business requirement for integrating a new analytics tool:

Specify the needs and goals for adding analytics capabilities.

Quickly create draft acceptance criteria to ensure that the feature described in the BRD is clearly defined, measurable, and mutually understood among stakeholders:

Create acceptance criteria for the feature outlined as: <paste in BRD text that describes the feature>

Visual Representations of Information

Visualise the overall structure and components of the repository:

Create a mermaid diagram of the architecture for this repository.

Understand the overall flow of this application through diagrams:

Visualise the flow of this application until the bot responds to a message using mermaidJS notation.

Create a mermaid diagram of the Business Requirement Document plan:

Illustrate the plan for the BRD.

Show the timeline for implementing the BRD:

Create a mermaid diagram of the Business Requirements Document timeline plan.

