

# Enhanced Documentation Generator

## Concept:

- Develop a tool that scans the entire codebase, extracts information from code annotations and structure, and generates comprehensive documentation for all functions. This tool could integrate with code repositories and update documentation automatically with each code change.
- The generated documentation will include details such as:
  - Function name
  - Description
  - Parameters
  - Return type
  - Exceptions
  - Usage examples

## Use:

The Enhanced Documentation Generator scans the codebase to extract and compile information from code annotations and docstrings, producing detailed, formatted documentation for all functions. This documentation, including descriptions, parameters, return types, exceptions, and usage examples, allows developers to quickly reference existing functions before writing new ones, ensuring code reuse and consistency. It can also be integrated with an LLM for efficient querying by developers, streamlining the development process.

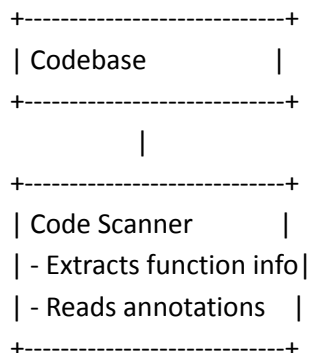
## Implementation Flow:

### 1. Code Scanning:

- Scan the entire codebase for functions and methods.
- Extract information from code annotations and docstrings.

### 2. Documentation Generation:

- Use a template to format the extracted information into readable documentation.
- Include sections for function description, parameters, return types, exceptions, and usage examples.



|

+-----+	
Documentation Generator	
- Formats info into docs	
- Adds descriptions	
- Creates usage examples	
+-----+	