Example Documentation generation

Repository Documentation

Repository: https://github.com/Pyro-Warrior-1884/Code-Documentation-Generator

Scanned path: C:\Users\Albert Nedumudy\Desktop\Main HQ\Work\DocuGen\_Run\Code\_Repository

Generated by: docgen.py (Ollama backend)

# Dependencies & Top-level Files

No dependency files were found.

# File Summaries

## doc\_gen\_script.py

## Combined Summary for `file\_gen\_script.py`  
  
\*\*Purpose:\*\* This script generates comprehensive developer documentation for a GitHub repository using Ollama.  
  
\*\*Key Components:\*\*  
  
\* `find\_code\_files\_bfs` function: recursively iterates through the repo tree and identifies code files.  
\* `read\_file` function: reads the content of a file and returns it as a string.  
\* `ollama\_generate` function: interacts with the Ollama API to generate a documentation summary from a given prompt.  
\* `chunk\_text` function: splits the input text into chunks of the specified size.  
\* `write\_docx` function: takes the output path, repo root, and file summaries as input and generates a Word document with the documentation.  
  
\*\*Dependencies:\*\*  
  
\* `GitPython`  
\* `python-docx`  
\* `requests`  
  
\*\*Edge Cases:\*\*  
  
\* The script may fail to find code files if the repo is not properly initialized.  
\* The output Word document may be empty or contain binary files.  
  
\*\*Additional Notes:\*\*  
  
\* The script assumes the presence of a `requirements.txt` file in the repo.  
\* Users can specify the output file path using the `--out` argument.  
\* The script will keep the cloned repo at the specified `repo\_root` directory.  
\* The `--keep` flag removes the temporary clone directory after the documentation generation.