Documentation-Generator-Agent — Project Documentation Report

1. Introduction

The Documentation-Generator-Agent is an AI-powered system designed to automatically analyze Python codebases and generate structured, readable Markdown documentation. The system reduces manual documentation effort by understanding code structure, dependencies, and semantics using Large Language Models (LLMs).

This project was developed as part of the **Gen AI for Gen Z – Intel® Unnati Industrial Training Spring 2026** initiative.

---

2. Problem Statement

Manual documentation of software projects is time-consuming, inconsistent, and often neglected. Developers focus on writing functional code while documentation becomes outdated or incomplete.

There is a need for an automated system that: - Understands project structure - Identifies modules, functions, and dependencies - Generates clear Markdown documentation automatically

---

3. Objectives

- Automatically parse Python projects
- Build dependency understanding between modules
- Generate structured Markdown documentation
- Provide LLM-based explanations for code
- Support both local projects and GitHub repositories
- Offer a simple web interface for user interaction

---

4. System Architecture

The system consists of the following major components:

1. Code Parser — Extracts functions, classes, imports, and structure
2. Dependency Analyzer — Builds relationships between files and modules
3. LLM Documentation Engine — Generates human-readable explanations
4. Markdown Generator — Formats output into clean documentation
5. Web Interface (optional) — Allows browser-based execution

5. Technologies Used

- Python
- LLM APIs (Gemini / ScaleDown AI)
- Flask / Web framework for UI
- Markdown generation
- AST (Abstract Syntax Tree) for code parsing

---

6. Working Methodology

1. User provides a Python project path or GitHub repo URL
2. System parses all Python files using AST
3. Dependencies and structure are mapped
4. Relevant code chunks are sent to LLM
5. LLM returns explanations
6. Markdown documentation is generated automatically

---

7. Key Features

- Automatic Python code analysis
- Dependency graph understanding
- LLM-powered documentation writing
- GitHub repository support
- Optional browser-based interface
- Structured Markdown output

---

8. Output Format

The generated documentation includes: - Project overview - File-wise explanations - Function and class descriptions - Dependency explanations

---

9. Advantages

- Saves developer time
- Ensures consistent documentation
- Useful for students and teams
- Easy to integrate into workflows

---

10. Limitations

- Currently focused on Python projects
- Depends on LLM quality for explanations
- Requires API keys and internet access

---

11. Future Enhancements

- Support for multiple languages
- Visual dependency graphs
- CI/CD integration
- PDF/HTML export

---

12. Conclusion

The Documentation-Generator-Agent demonstrates how Generative AI can automate essential but neglected tasks like documentation. It improves productivity and helps developers maintain high-quality project records with minimal effort.