AutoDoc – Autonomous Project Documentation & Diagram Generator

(Agentic AI Tool for Automatic Documentation)

🎯 1. Project Summary

AutoDoc automatically:

- Reads your GitHub repository
- Understands source code
- Generates documentation (README, API docs, flow diagrams, class diagrams, ER diagrams, architecture diagrams)
- Maintains versioned documentation
- Updates docs automatically on every commit using GitHub Actions
- Uses LLM agents to analyze code & produce accurate documentation

This is a **highly practical** Agentic AI project.

🧠 2. Key Features of AutoDoc

✓ A. Codebase Analyzer Agent

- Reads all files: .py, .java, .js, .ts, .md, .html, .css, etc.
- Extracts class definitions, functions, modules, APIs.
- Creates a full understanding graph of the project.

✓ B. Documentation Generator Agent

Automatically generates:

- **README.md**
- **Project overview**
- **API Documentation**
- **Backend + Frontend Flow**
- **Class Diagrams (UML)**

- **Sequence Diagrams**
- **Architecture Diagram**
- **Environment Setup Guide**
- **Deployment Guide**
- **System Design Document**

✓ C. Diagram Generator Agent

Converts AI output into:

- Mermaid diagrams
- UML diagrams
- Flowcharts
- Component diagrams
- ER Diagrams (DB)

✓ D. GitHub Integration Agent (Fully automatic)

Automatically runs when new code is pushed.

Workflow:

1. User pushes code → GitHub
2. GitHub Actions triggers the AutoDoc engine
3. AutoDoc reads changed files
4. AutoDoc updates documentation
5. AutoDoc commits the new documentation back into repo
6. Sends developer a summary email or GitHub notification

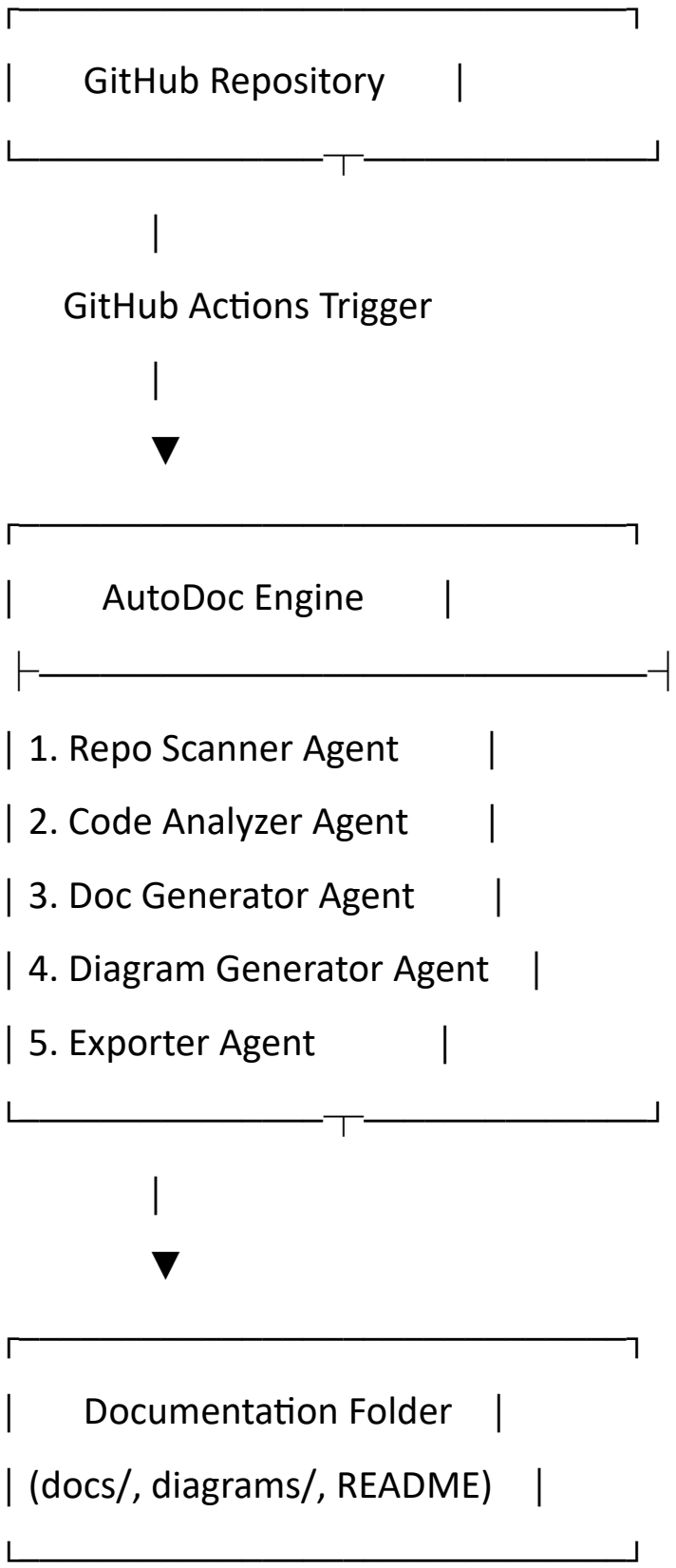
✓ E. Document Exporting Engine

Exports documentation to:

- PDF

- Markdown
- HTML
- JSON API

🏗️ 3. System Architecture



🛠️ 4. Tech Stack

Backend

- **Python + FastAPI**
  OR
- **Node.js + Express**

AI/LLM Tools

- OpenAI GPT-4.1 / GPT-o-mini
- LangChain
- Embeddings for code understanding
- Agentic AI pipeline

DevOps

- GitHub Actions
- Docker (optional)

Frontend Dashboard (optional but recommended)

- React.js
- Tailwind CSS
- ShadCN UI

📦 5. Core Modules (High-Level)

1. Repo Scanner

- Clones GitHub repo locally
- Reads file tree
- Detects programming languages
- Extracts relevant files

2. Code Understanding Agent

Uses LLM to interpret:

- Classes
- Functions
- APIs
- Endpoints
- Database models
- Microservice connections
- Frontend → backend communication

3. Documentation Generator

Outputs multiple documents like:

- System Overview Summary
- README
- API Docs
- UML Diagrams
- Flowcharts
- DB Diagram
- Overall Architecture Diagram

4. Diagram Creator

Converts text → Mermaid diagrams, example:

graph TD

A[Frontend] --> B[Backend API]

B --> C[Database]

5. Auto Commit Agent

Commits generated docs to GitHub automatically.

🔌 6. GitHub Integration (How it Works)

Step 1: Developer pushes code

Step 2: GitHub Action runs:

```
name: AutoDoc

on:
  push:
    branches: [ "main" ]

jobs:
  run-autodoc:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout Repo
        uses: actions/checkout@v3

      - name: Run AutoDoc
        run: |
          pip install autodoc-engine
          python run_autodoc.py

      - name: Commit Documentation
        run: |
          git config user.name "AutoDoc Bot"
          git config user.email "autodoc@bot.com"
          git add .
          git commit -m "Auto-generated documentation update"
          git push
```

Step 3: AutoDoc creates/updates documentation

Step 4: Commits documentation

Step 5: Developer sees updated files automatically

🎓 7. Why AutoDoc is a STRONG Project?

✓ High Industry Relevance

Every developer faces documentation problems.

✓ Agentic AI + DevOps Integration

Very rare among students.

✓ Shows you understand:

- AI
- Backend
- GitHub Actions
- Automation
- Diagrams
- Full-stack tools

✓ Unique, modern, practical.