# Django Project Documentation Tools

## Table of Contents

# 1. What is Documentation?

Documentation is a detailed written description that explains how a project or software works. It provides guidelines for installation, setup, configuration, and usage, ensuring that other developers can easily understand, maintain, and extend your Django project.

# 2. Why Do We Need Documentation?

- To help new developers understand the project structure and logic.
- To ensure consistency and maintainability of the codebase.
- To provide users with clear setup and usage instructions.
- To serve as a reference guide for troubleshooting and upgrading.
- To improve collaboration among development teams.

# 3. Documentation Tools Overview

Two popular documentation tools suitable for Django projects that are

> - Sphinx
> - MsDoc

# 4. Sphinx Documentation Tool

Sphinx is a powerful documentation generator originally created for Python projects. It uses reStructuredText or Markdown as source files and can generate HTML, PDF, or ePub outputs.

## 4.1. Requirements

Make sure you have:

- **Python** installed
- **pip** (Python package installer) working

You can check by opening **Command Prompt (CMD)** or **PowerShell** and typing:

```
python –version
pip --version
```

If these return version numbers (e.g., Python 3.12.2 and pip 24.0), you're good to go.
If not, first install Python.

## 4.2. Installation

To install Sphinx, run the following command:

```
pip install -u sphinx
```

## 4.3. Creating the documentation layout and Build

Then from the command line, run the following command:

```
sphinx-quickstart docs
```

This will present to you a series of questions required to create the basic directory and configuration layout for your project inside the docs folder. To proceed, answer each question as follows:

```
>Separate source and build directories (y/n) Write: y and press Enter.
>Project name: Write "Kenakata" and press Enter.
>Author names: Write "Rishat" and press Enter.
>Project release: Write "0.1" (without quotes) and press Enter.
>Project Language: Leave it empty (the default, English) and press Enter.
```

After the last question, you will see the new docs directory with the following content.

```
docs
├── build
├── make.bat
├── Makefile
└── source
    ├── conf.py
    ├── index.rst
    ├── static
    └── templates
```

After setup, you can build the HTML docs with this command:

```
sphinx-build -b html source build
```

And finally, open docs/build/html/index.html in your browser. You should see something like this:

There we go! You created your first HTML documentation using Sphinx.

## 4.4. Advantages

- Supports multiple output formats (HTML, PDF, ePub).
- Integrates well with Python/Django.
- Auto-generates documentation from docstrings.
- Supports theming and extensions like ReadTheDocs.

## 4.5. Disadvantages

- Has a learning curve for reStructuredText syntax.
- Configuration files can be complex for beginners.
- May require extra setup for Markdown support.

# 5. MkDocs Documentation Tool

MkDocs is an open-source static site generator that is specifically designed for building project documentation. It converts simple Markdown (.md) files into a beautiful and searchable website — all from the command line, with minimal setup. It's written in Python, which makes it easy to install and use for Django and other Python-based projects.

## 5.1. Requirements

Make sure you have:

- **Python** installed
- **pip** (Python package installer) working

You can check by opening **Command Prompt (CMD)** or **PowerShell** and typing:

```
python –version
```

```
pip --version
```

If these return version numbers (e.g., Python 3.12.2 and pip 24.0), you're good to go.
If not, first install Python.

## 5.2. Installation

To install Sphinx, run the following command:

```
pip install mkdocs
```

## 5.3. Creating the documentation layout and Build

In your Django project run the commad:

```
mkdocs new docs
cd docs
```

this will create a folder structure like this:

```
docs/
├──── docs/
│      └──── index.md
└──── mkdocs.yml
```

Now edit your documents file **docs/index**.md looking like this:

```
# Welcome to My Django Project Docs

This documentation describes how to set up, configure, and
run the Django project.

## Installation
1. Install Python and Django
2. Run `python manage.py runserver`

## Features
- User Authentication
- Admin Panel
- REST API Integration
```

For the next step you need to configure **mkdocs.yml** for that open mkdocs.yml and edit it like :

```
site_name: My Django Documentation
theme:
  name: material
nav:
  - Home: index.md
  - Setup: setup.md
  - API: api.md
```

Now build static HTML files for finalize the documentation run command :

```
mkdocs build
```

This creates a folder named site/ containing static HTML files Like this :

```
docs/
├──── docs/
├──── mkdocs.yml
└──── site/
      ├──── index.html
      ├──── setup/index.html
      └──── api/index.html
```

And done this is your first documentation for mkdocs.

## 5.4. Advantages

- Easy to install and set up
- Uses simple Markdown syntax
- Fast and lightweight documentation generator
- Supports beautiful themes like Material
- Easy to deploy on GitHub Pages or any server

## 5.5. Disadvantages

- Does not auto-generate docs from Python code by default
- Limited plugin and extension support
- Markdown offers fewer formatting options than re Structured Text
- Manual linking required between pages
- Not ideal for very large or complex documentation projects