

Documentation tool for an E Commerce. website(Elektra.)

Sphinx

Since we are using **Python** and **Django** for our project (**Elektra- an E-Commerce website**) , we choose **Sphinx** ;a powerful and widely used documentation generation tool for **Python Projects**. Originally created for the Python documentation itself , it is now used by many open source and commercial projects to generate high quality technical documentation because It is designed to make software documentation easier to write, manage and navigate. It supports formats like **reStructuredText** and **Markdown**, giving us the flexibility to structure our documentation the way we need.

With Sphinx, We can keep our documentation well organized, version controlled and easy to update. This ensures that everything from **installation guides** to **feature explanations** and **system workflows** is clearly documented, whether it's for developers working on the back end or users trying to understand the platform, Sphinx helps us keep things accessible and structured.

Key Features:

- **Automatic API Documentation:** Extracts docstrings from Python code.
- **Multiple Output Formats:** supports HTML, Latex(PDF) EPUB and more.
- **Extensions & Theming:** Customizable via themes and plugins.
- **Integration with ReadTheDocs:** Easily host and update documentation.

Code Documentation

To ensure technical clarity, we integrated Sphinx autodoc to extract docstrings from our Python based backend. This enabled automatic generation of reference documentation for the following system components.

- **User access management.**
- **Product inventory system.**

- Order handling mechanism.
- Secure payment processing.

Installation & Usage

1. installation sphinx : python 3.13.2

```
Select Command Prompt

Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ai>python --version
Python 3.13.2
```

Installation link Sphinx :

(<https://www.sphinx-doc.org/en/master/usage/installation.html>)

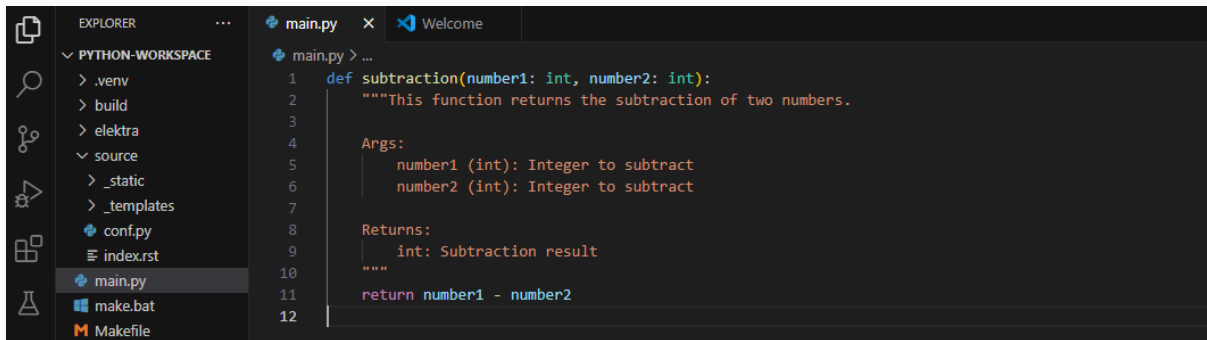
2. Source Code:

```
EXPLORER
PYTHON-WORKSPACE
> .venv
> build
> elektra
> source
  > _static
  > _templates
  conf.py
  index.rst
  main.py
  make.bat
  Makefile

main.py
1 def main():
2     """
3     this function prints a message from main functions.
4     """
5     print ("Hello from main.py!")
6
7     if __name__=="__main__":
8         main()
9
10
```

```
EXPLORER
PYTHON-WORKSPACE
> .venv
> build
> elektra
> source
  > _static
  > _templates
  conf.py
  index.rst
  main.py
  make.bat
  Makefile

main.py
1 def summation(number1: int, number2: int):
2     """This function returns the summation of two numbers.
3
4     Args:
5         number1 (int): Integer to sum
6         number2 (int): Integer to sum
7
8     Returns:
9         int: Summation result
10    """
11    return number1 + number2
12
```



Why Use Sphinx?

1. Automated Documentation

- extracts docstrings from Python code to generate API documentation automatically ; reduces manual effort in keeping docs up to date.

2. Multiple Output Formats

- Generate HTML, PDF(LaTex), EPUB and more from a single source.

3. Customization & Theming

- Supports different themes and extensions that allows advanced features like search, diagrams and interactive examples;so that we can make our docs look polished and branded.

4. ReStructuredText (reST) Support

- A markup language similar to Markdown, but more powerful for technical docs and supports tables, code blocks and references easily.

5. Great for Large Projects

- Handles multi page documentation, cross references and indexing effortlessly.
- Many popular projects like Django and NumPy use Sphinx

Documentation Deliverables

After compilation, the final structured documentation includes:

- user instructions with step by step explanation.
- System design overview for developers.
- API documentation for back end integrations.
- Help and support section addressing common queries.
- system requirements and compatibility for supported browsers, platforms and devices

Benefits & Limitations of Sphinx

Advantages of using Sphinx

- **Multiple output formats** – HTML, PDF, ePub, etc.
- **Automated documentation generation** from structured text.
- **Built-in cross-referencing & indexing** for easy navigation.
- **Version control compatibility** with Git for long-term maintainability.

Disadvantages of using Sphinx

- **Learning Curve** – Requires familiarity with reStructuredText, which may not be as intuitive as Markdown.
- **Setup Complexity** – Initial configuration can be challenging, especially for beginners
- **Customization Limitations** – Deep customization requires extra effort despite theming options.
- **Dependency on Python** – Mainly designed for Python projects, limiting flexibility for other stacks.

By utilizing **Sphinx**, we have successfully documented our ecommerce website, ensuring that developers, vendors and customers can access well organized and comprehensive documentation to support future growth and updates.

