Documentation Tool

The Bengali Newspaper project will use the Django framework. As the Django framework is based on Python, we need to use a document generator based on Python. We will be using Sphinx as our documentation generator.

**Sphinx**

Sphinx, written by Georg Brandl and licensed under the BSD license, was originally created for the Python documentation and it has excellent facilities for the documentation of software projects in a range of languages. It's written in C++ and works on Linux and can work on many known operating system. Django’s documentation is intended to be read as HTML.

**Sphinx Features**

* Batch and Real-Time full-text indexes. Two index backends that support both efficient offline index construction and incremental on-the-fly index updates are available.
* Non-text attributes support. An arbitrary number of attributes (product IDs, company names, prices, etc) can be stored in the index and used either just for retrieval (to avoid hitting the DB), or for efficient Sphinx-side search result set post-processing.
* SQL database indexing. Sphinx can directly access and index data stored in MySQL (all storage engines are supported), PostgreSQL, Oracle, Microsoft SQL Server, SQLite, Drizzle, and anything else that supports ODBC.
* Non-SQL storage indexing. Data can also be streamed to the batch indexer in a simple XML format called XMLpipe, or inserted directly into an incremental RT index.
* Easy application integration. Sphinx comes with three different APIs, SphinxAPI, SphinxSE, and SphinxQL. SphinxAPI is a native library available for Java, PHP, Python, Perl, C, and other languages. SphinxSE, a pluggable storage engine for MySQL, enables huge result sets to be shipped directly to MySQL server for post-processing. SphinxQL lets the application query Sphinx using standard MySQL client libary and query syntax.
* Advanced full-text searching syntax. Our querying engine supports arbitrarily complex queries combining boolean operators, phrase, proximity, strict order, and quorum matching, field and position limits, exact keyword form matching, substring searches, etc.
* Rich database-like querying features. Sphinx does not limit you to just keyword searching. On top of a full-text search result set, you can compute arbitrary arithmetic expressions, add WHERE conditions, do ORDER BY, GROUP BY, use MIN/MAX/AVG/SUM, aggregates etc. Essentially, full-blown SQL SELECT is supported.
* Better relevance ranking. Unlike many other engines, Sphinx does not solely rely on 30-year-old statistical ranking that only considers keyword frequencies, nor limits you to it. By default, Sphinx additionally analyzes keyword proximity, and ranks closer phrase matches higher, with perfect matches ranked on top. Also, ranking is flexible: you can choose from a number of built-in relevance functions, tweak their weights by using expressions, or develop new ones.
* Flexible text processing. Sphinx indexing features include full support for SBCS and UTF-8 encodings (meaning that effectively all world's languages are supported); stopword removal and optional hit position removal (hitless indexing); morphology and synonym processing through word forms dictionaries and stemmers; exceptions and blended characters; and many more.
* Distributed searching. Searches can be distributed across multiple machines, enabling horizontal scale-out and HA (High Availability).

**Pros**

* Documents are written as plain text files.
* Output formats will be generated by Sphinx: HTML, PDF, eBook.
* Markup language used (REST) is similar to Wiki languages (Markdown, Textile, ...).
* Layouts can be replaced, many default layouts available.
* Writer concentrates on content, layout and output comes from Sphinx.
* Autogenerating documentation from source code.
* Syntax highlighting for many formats (Python, XML, ...).
* Cross-references to parts of the documentation (sections, code, glossary, ...) and automatic indices.
* Version control (GitHub) can be used for collaborative work and keeping track of different document versions.
* The documentation is part of the source code repository.

**Cons**

* Has somewhat external dependency.
* Might be difficult to set-up as it is not an API generation documentation tool.
* Since it’s not an API documentation tool, generating call/inheritance graphs might be difficult.

**Download Sphinx**

[Download Sphinx](http://www.sphinxsearch.com/)

**Installation Process:**

You can install Sphinx from PyPI with pip:

$ pip install sphinx

Or install it with Anaconda:

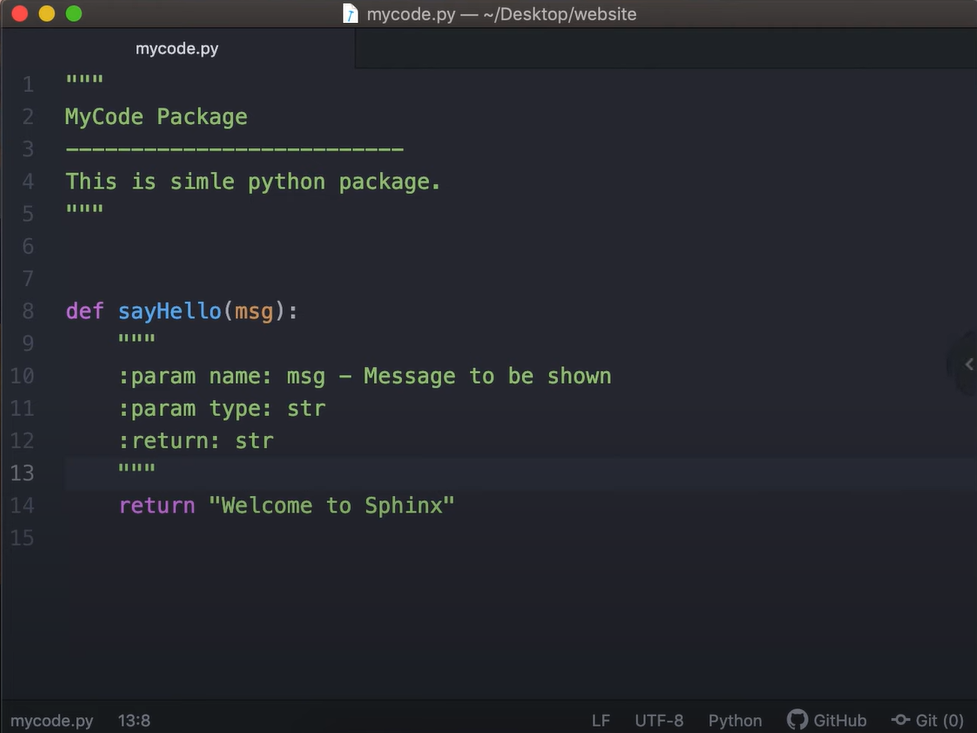
$ conda install sphinx

Link on Installation Process: [Install Sphinx](https://youtu.be/8mDDEStuBes)

**Using Sphinx:**

Link on Sphinx Setup: [Sphinx Set-Up](https://youtu.be/MeaDUypDAoI)

**Source Code Documentation Example:**



### HTML Documentation:

You can know more about sphinx: [Quick Sphinx Tutorial](https://quick-sphinx-tutorial.readthedocs.io/en/latest/index.html)

More Useful Links:

[About Sphinx](http://www.sphinxsearch.com/about/sphinx/)

[Document Project Using Sphinx](https://medium.com/@richdayandnight/a-simple-tutorial-on-how-to-document-your-python-project-using-sphinx-and-rinohtype-177c22a15b5b)