File: .pdfignore.yml

```
directories:
    - .git
    - log
    - public/system
    - tmp
    - vendor
    - build
    - dist
files:
    - .DS_Store
    - database.yml
    - favicon.ico
patterns:
    - '*.pdf'
    - '*.tmp'
    # - '*.md'
    # - '*.py'
```

File: requirements.txt

```
Markdown==3.6
PyYAML==6.0.1
reportlab==4.2.0
```

File: code-to-pdf.py

```
import os
import yaml
import fnmatch
from reportlab.lib.pagesizes import letter
from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer
from reportlab.lib.styles import getSampleStyleSheet, ParagraphStyle
from reportlab.pdfbase import pdfmetrics
from reportlab.pdfbase.cidfonts import UnicodeCIDFont
code_extensions = [
    ".py", ".js", ".html", ".css", ".php", ".java", ".cpp", ".c",
    ".h", ".hpp", ".cs", ".rb", ".pl", ".sh", ".sql", ".json",
    ".xml", ".yaml", ".yml", ".md", ".rst", ".txt"
]
def load_exclusions(config_path):
    with open(config_path, 'r', encoding='utf-8') as file:
        config = yaml.safe_load(file)
    excluded_dirs = set(config.get('directories', []))
    excluded_files = set(config.get('files', []))
    excluded_patterns = config.get('patterns', [])
    return excluded_dirs, excluded_files, excluded_patterns
def generate_pdf(source_folder, output_pdf, config_path):
    excluded_dirs, excluded_files, excluded_patterns = load_exclusions(config_path)
    doc = SimpleDocTemplate(output_pdf, pagesize=letter)
    story = []
    styles = getSampleStyleSheet()
    pdfmetrics.registerFont(UnicodeCIDFont('STSong-Light'))
    code_style = ParagraphStyle('Code', parent=styles['Normal'], fontName='STSong-Light', fontSize=10, leading=12)
    for root, dirs, files in os.walk(source_folder, topdown=True):
        dirs[:] = [d for d in dirs if d not in excluded_dirs]
        for filename in files:
            if any(filename.endswith(ext) for ext in code_extensions):
                if filename not in excluded_files and not any(fnmatch.fnmatch(filename, pattern) for pattern in excluded_patterns):
                    if not any(root.startswith(os.path.join(source_folder, d)) for d in excluded_dirs):
                        filepath = os.path.join(root, filename)
                        with open(filepath, 'r', encoding='utf-8') as file:
                            code = file.read()
                            story.append(Paragraph(f"File: {filename}", styles['Title']))

                            for line in code.splitlines():
                                story.append(Paragraph(line.replace(' ', ' '), code_style))

                            story.append(Spacer(1, 12))

    doc.build(story)
# 使用该函数
generate_pdf('./', 'Code.pdf', '.pdfignore.yml')
```

File: README.md

# Code to PDF Converter 代码转换成PDF工具

Code to PDF Converter is a Python tool that converts source code files in a directory to a single PDF document, excluding files and directories specified in a configuration file. This tool is especially useful for developers who need to archive or share their codebase in a print-friendly format.

代码转换成 PDF 工具 是一个 Python 工具，用于将目录中的源代码文件转换为单个 PDF 文档，同时可以通过配置文件指定排除某些文件和目录。该工具对需要以打印友好格式存档或分享代码库的开发者尤其有用。

## Features 功能

- Convert all source code files in a directory to a single PDF. 将一个目录下的所有源代码文件转换成一个 PDF 文件。
- Exclude specific files and directories using a configuration file. 使用配置文件排除特定的文件和目录。
- Support for custom file patterns via glob patterns in the configuration. 通过配置中的通配符模式支持自定义文件模式。

## Installation 安装指南

To use this tool, you need Python installed on your system. Clone this repository and install the required dependencies:

要使用此工具，您需要在系统上安装 Python。克隆此仓库并安装所需的依赖项：

```bash
git clone git@github.com:johnsmzr/Code-to-one-PDF.git
```

cd Code-to-one-PDF
pip install -r requirements.txt
~~~
## Usage 使用说明
To generate a PDF from your code files, run the script from the command line:
要从代码文件生成 PDF，请在命令行运行脚本：
bash
Copy code
`python code-to-pdf.py -s /path/to/source -o /path/to/output.pdf -c /path/to/config.yml`
### Command Line Arguments 命令行参数
- `-s` or `--source`: Source directory containing the code files.

- `-o` or `--output`: Output path for the generated PDF.

- `-c` or `--config`: Path to the configuration file.

- `-s` 或 `--source`：包含代码文件的源目录。

- `-o` 或 `--output`：生成的 PDF 的输出路径。

- `-c` 或 `--config`：配置文件的路径。

## Configuration File 配置文件说明
The configuration file `.code2pdf.yml` should be placed in the root of the source directory. It specifies which files and directories to exclude. Example:
配置文件 `.code2pdf.yml` 应放置在源目录的根目录中。它指定了要排除哪些文件和目录。例如：
```yaml
directories:
    - .git
    - log
files:
    - .DS_Store
patterns:
    - '*.pdf'
    - '*.tmp'
~~~
## FAQ 常见问题解答
Q: Can I exclude files based on their content?
A: Currently, the exclusion is based only on file names and directory paths specified in the config file.
Q: 我可以根据文件内容排除文件吗？
A: 目前，排除仅基于配置文件中指定的文件名和目录路径。
