./.pdfignore.yml

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

./requirements.txt

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

./code-to-pdf.py

```
import os
import yaml
import fnmatch
from fpdf import FPDF

# Load the blacklist from the .code2pdf file
blacklist_file = '.pdfignore.yml'
if os.path.exists(blacklist_file):
    with open(blacklist_file, 'r') as f:
        blacklist = yaml.safe_load(f)
else:
    blacklist = {'directories': [], 'files': [], 'patterns': []}

# Define file extensions to consider as code files
code_extensions = ['.py', '.js', '.html', '.css', '.php', '.java', '.cpp', '.c', \
    '.h', '.hpp', '.cs', '.rb', '.pl', '.sh', '.sql', '.json', '.xml', '.yaml', '.yml', \
    '.md', '.rst', '.txt']

# Create a new PDF object
pdf = FPDF()

# Function to add a file to the PDF
def add_file_to_pdf(file_path):
    with open(file_path, 'r', encoding='utf-8') as f:
        content = f.read()
        for line in content.split('\n'):
            # pdf.add_font('courier', '', 'courier.ttf', uni=True)
            # pdf.add_font('Courier', '', None, uni=True)
            pdf.set_font('Arial', '', 10)
            line = line.encode('latin-1', 'replace').decode('latin-1')
            pdf.cell(0, 5, txt=line, ln=1)

# Iterate over all files and directories in the current directory
for root, dirs, files in os.walk('.'):
    dirs[:] = [d for d in dirs if d not in blacklist['directories']]
    for file in files:
        if (file not in blacklist['files']
            and os.path.splitext(file)[1].lower() in code_extensions
            and not any(fnmatch.fnmatch(file, pattern) for pattern in blacklist['patterns'])):

            file_path = os.path.join(root, file)
            pdf.add_page()
            pdf.set_font('courier', 'B', 16)
            pdf.cell(0, 10, txt=file_path, ln=1)
            add_file_to_pdf(file_path)

# Save the PDF file
pdf.output('code.pdf', 'F')
print('Code files have been converted to code.pdf')
```

./test.py

```
import os
from reportlab.lib.pagesizes import letter
from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer
from reportlab.lib.styles import getSampleStyleSheet, ParagraphStyle

def generate_pdf(source_folder, output_pdf):
    doc = SimpleDocTemplate(output_pdf, pagesize=letter)
    story = []
    styles = getSampleStyleSheet()
    code_style = ParagraphStyle('Code', parent=styles['Normal'], fontName='Courier', fontSize=10, leading=12)

    for filename in sorted(os.listdir(source_folder)):
        if filename.endswith(".py"):  # ??????????
            filepath = os.path.join(source_folder, 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(' ', '&nbsp;'), code_style))
                story.append(Spacer(1, 12))

    doc.build(story)

# ?????
generate_pdf('./', 'output_code2.pdf')
```

./README.md

# Code to PDF Converter ?????PDF??

Code to PDF Converter is a Python tool that converts source code files in a directory to a single PDF document, excluding fil

????? PDF ?? ??? Python ???????????????????? PDF ???

## Features ??

- Convert all source code files in a directory to a single PDF. ??????????????????? PDF ???
- Exclude specific files and directories using a configuration file. ?????????????????
- Support for custom file patterns via glob patterns in the configuration. ?????????????????????

## Installation ????

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

???????????????? Python????????????????

```bash
git clone git@github.com:johnsmzr/Code-to-one-PDF.git
cd Code-to-one-PDF
pip install -r requirements.txt
```

## Usage ????

To generate a PDF from your code files, run the script from the command line:

???????? PDF???????????

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 ?????

- `-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` ? `--source`????????????

- `-o` ? `--output`???? PDF ??????

- `-c` ? `--config`?????????

## Configuration File ??????

The configuration file `.code2pdf.yml` should be placed in the root of the source directory. It specifies which files and directories

???? `.code2pdf.yml` ??????????????????????????????

```yaml
directories:
  - .git
  - log
files:
  - .DS_Store
patterns:
  - '*.pdf'
  - '*.tmp'

```

## FAQ ??????

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: ???????????????

A: ?????????????????????????
