

```
--- project_structure.py ---
```

```
import os
```

```
def list_files_and_folders(directory, output_file):
```

```
    with open(output_file, 'w') as f:
```

```
        for root, dirs, files in os.walk(directory):
```

```
            # Ignore .git and __pycache__ folders
```

```
            dirs[:] = [d for d in dirs if d not in ['.git', '__pycache__']]
```

```
            f.write(f"Directory: {root}\n")
```

```
            for dir_name in dirs:
```

```
                f.write(f"  Folder: {dir_name}\n")
```

```
            for file_name in files:
```

```
                f.write(f"    File: {file_name}\n")
```

```
# Update the directory path to your project folder
```

```
project_directory = "D:/HARRISBURG/Harrisburg Master's Fifth Term Late Summer/CISC  
699/DiscordBotProject_CISC699"
```

```
output_file = os.path.join(project_directory, "project_structure.txt")
```

```
# Call the function to list files and save output to .txt
```

```
list_files_and_folders(project_directory, output_file)
```

```
print(f"File structure saved to {output_file}")
```

```
--- project_text.py ---
```

```
import os
```

```
from fpdf import FPDF
```

```
# Directory where the project files are located
```

```
directory = r"D:\HARRISBURG\Harrisburg Master's Fifth Term Late Summer\CISC  
699\DiscordBotProject_CISC699\other"
```

```
output_pdf_path = os.path.join(directory, "project_text.pdf")
```

```
# Function to retrieve all text from files, ignoring .git and __pycache__ directories
```

```
def extract_project_text(directory):
```

```
    project_text = ""
```

```
    for root, dirs, files in os.walk(directory):
```

```
        # Ignore .git and __pycache__ directories
```

```
        dirs[:] = [d for d in dirs if d not in ['.git', '__pycache__']]
```

```
        for file in files:
```

```
            if file.endswith('.py') or file.endswith('.txt') or file.endswith('.md'): # Only considering relevant
```

```
file types
```

```
            file_path = os.path.join(root, file)
```

```
            try:
```

```
                with open(file_path, 'r', encoding='utf-8') as f:
```

```
                    project_text += f"--- {file} ---\n"
```

```
                    project_text += f.read() + "\n\n"
```

```
            except Exception as e:
```

```
                print(f"Could not read file {file_path}: {e}")
```

```
    return project_text
```

```
# Function to generate a PDF with the extracted text
```

```
def create_pdf(text, output_path):
```

```
    pdf = FPDF()
```

```
    pdf.set_auto_page_break(auto=True, margin=15)
```

```
    pdf.add_page()
```

```
    pdf.set_font("Arial", size=12)
```

```
    # Ensure proper encoding handling
```

```
    for line in text.split("\n"):
```

```
        # Convert the text to UTF-8 and handle unsupported characters
```

```
        try:
```

```
            pdf.multi_cell(0, 10, line.encode('latin1', 'replace').decode('latin1'))
```

```
        except UnicodeEncodeError:
```

```
            # Handle any other encoding issues
```

```
            pdf.multi_cell(0, 10, line.encode('ascii', 'replace').decode('ascii'))
```

```
    pdf.output(output_path)
```

```
# Extract project text and create the PDF
```

```
project_text = extract_project_text(directory)
```

```
if project_text:
```

```
    create_pdf(project_text, output_pdf_path)
```

```
    output_pdf_path
```

```
    print("PDF file created with all project's as text at: " + output_pdf_path)
```

else:

"No project text found."

--- Tests\_URLs.txt ---

database password: postgres

Working Commands: Test commands

!project\_help

!login bestbuy

!launch\_browser

!close\_browser

!navigate\_to\_website <https://www.google.com/>

!get\_price

<https://www.bestbuy.com/site/microsoft-xbox-wireless-controller-for-xbox-series-x-xbox-series-s-xbox-one-windows-devices-sky-cipher-special-edition/6584960.p?skuld=6584960>

!monitor\_price

<https://www.bestbuy.com/site/microsoft-xbox-wireless-controller-for-xbox-series-x-xbox-series-s-xbox-one-windows-devices-sky-cipher-special-edition/6584960.p?skuld=6584960>

x-one-windows-devices-sky-cipher-special-edition/6584960.p?skuId=6584960

!stop\_monitoring

!check\_availability <https://www.opentable.com/r/bar-spero-washington/>

!monitor\_availability <https://www.opentable.com/r/bar-spero-washington/>

!monitor\_availability <https://www.opentable.com/r/bar-spero-washington/> "October 2"

!stop\_monitoring\_availability

!check\_availability <https://www.opentable.com/r/bar-spero-washington/> "October 2"

!stop\_bot

\*\*\*\*\*

Working on it:

!check\_availability <https://www.opentable.com/r/bar-spero-washington/> "August 22" "8:00 PM"

\*\*\*\*\*

URLs to Test:

<https://www.opentable.com/r/bar-spero-washington/>

[https://www.ebay.com/itm/314411766963?\\_trkparms=amclsrc%3DITM%26aid%3D777008%26algo%3DPERSONAL.TOPIC%26ao%3D1%26asc%3D20240603121456%26meid%3Da07931f944bc4a5b95376fe64d0ab035%26pid%3D102177%26rk%3D1%26rkt%3D1%26itm%3D314411766963%26pmt%3D1%26noa%3D1%26pg%3D4375194%26algv%3DNoSignalMostWatched%26brand%3DSimpliSafe&\\_trksid=p4375194.c102177.m166540&\\_trkparms=parentrq%3A71497a9c1910a8cd54f819a0ffff582e%7Cpageci%3A59d1354a-5f2b-11ef-9c4d-f2c982e61003%7Ciid%3A1%7Cvlpname%3Avlp\\_homepage](https://www.ebay.com/itm/314411766963?_trkparms=amclsrc%3DITM%26aid%3D777008%26algo%3DPERSONAL.TOPIC%26ao%3D1%26asc%3D20240603121456%26meid%3Da07931f944bc4a5b95376fe64d0ab035%26pid%3D102177%26rk%3D1%26rkt%3D1%26itm%3D314411766963%26pmt%3D1%26noa%3D1%26pg%3D4375194%26algv%3DNoSignalMostWatched%26brand%3DSimpliSafe&_trksid=p4375194.c102177.m166540&_trkparms=parentrq%3A71497a9c1910a8cd54f819a0ffff582e%7Cpageci%3A59d1354a-5f2b-11ef-9c4d-f2c982e61003%7Ciid%3A1%7Cvlpname%3Avlp_homepage)

<https://www.trendyol.com/puma/rebound-v6-low-p-736020132?boutiqueId=61&merchantId=184734&sav=true>

!get\_price

<https://www.trendyol.com/puma/rebound-v6-low-p-736020132?boutiqueId=61&merchantId=184734&sav=true>