Lab2 Brainstorm

Legal Advisory Chatbot

Members:

Zhizhuo Li 2248244297
Tsung Ting Lee

https://www.collegesoflaw.edu/blog/2018/11/06/different-fields-of-law-explained/

Domains:

Business Legal advisory(Tax Law, Corporate Law, Labor Law, Intellectual Property Law, Environmental Regulation, etc.)

Target Audiences: People who want to start a business

Most startups are facing the challenge of limited budgets for traditional legal services, which can make completing complex legal frameworks difficult and time-consuming. From incorporation laws and tax regulations to employment contracts and Intellectual Property Laws, startups must address a wide range of legal issues. This lack of affordable and accessible legal service resources can be a main barrier to the development and growth of the company. Our tailored business legal advisory solution will support your startups with AI-based automated tools, instant guidance, and structured references to avoid compliance failures. In addition, the business Legal chatbox generates incorporation documents, compliance checklists, and contract templates for startups to handle repetitive legal tasks assisting companies to focus on innovation while staying legally secure.

Data Sources:

Tax Law

Businesses of all sizes need guidance on tax compliance, deductions, and credits.
Startups often miss out on tax-saving opportunities due to lack of expertise

Corporate Law:

Entrepreneurs and startups need guidance on choosing the right business structure and complying with corporate governance laws.

Labor Law:

Labor law is critical for businesses to manage their workforce effectively while complying with regulations that protect employees' rights and establish employer obligations.

Banking Law:

Banking law assists startups in understanding fair loan agreements and ensuring credibility.

LLC Form 8832
https://www.irs.gov/pub/irs-pdf/f8832.pdf

https://www.irs.gov/businesses/business-tax-account

Form 940: Employer's Annual Federal Unemployment (FUTA) Tax Return
https://www.irs.gov/pub/irs-pdf/f940.pdf

Instruction for Form 940
https://www.irs.gov/pub/irs-pdf/i940.pdf

Form 943: Employer's Annual Federal Tax Return for Agricultural Employees
https://www.irs.gov/pub/irs-pdf/f943.pdf

Instruction for 943
https://www.irs.gov/pub/irs-pdf/i943.pdf

Form 944: Employer's ANNUAL Federal Tax Return
https://www.irs.gov/pub/irs-pdf/f944.pdf

Instruction for 944
https://www.irs.gov/pub/irs-pdf/i944.pdf

Form 945: Annual Return of Withheld Federal Income Tax
https://www.irs.gov/pub/irs-pdf/f945.pdf

Instruction for 945
https://www.irs.gov/pub/irs-pdf/i945.pdf

Form 941: Employer's QUARTERLY Federal Tax Return
https://www.irs.gov/pub/irs-pdf/f941.pdf

Instruction for 941
https://www.irs.gov/pub/irs-pdf/i941.pdf

Form 941X: Adjusted Employer’s QUARTERLY Federal Tax Return or Claim for Refund
https://www.irs.gov/pub/irs-pdf/f941x.pdf

Instruction for 941X
https://www.irs.gov/pub/irs-pdf/i941x.pdf

Form 943X: Adjusted Employer’s Annual Federal Tax Return for Agricultural Employees or Claim for Refund
https://www.irs.gov/forms-pubs/about-form-943-x

Instruction for 943X
https://www.irs.gov/pub/irs-pdf/i943x.pdf

Form 945X: Adjusted Annual Return of Withheld Federal Income Tax or Claim for Refund
https://www.irs.gov/pub/irs-pdf/f945x.pdf

Instruction for 945X
https://www.irs.gov/pub/irs-pdf/i945x.pdf

EIN:
https://www.irs.gov/businesses/employer-identification-number

Business Tax:
https://www.irs.gov/businesses/business-taxes

Records Keeping
https://www.irs.gov/businesses/small-businesses-self-employed/why-should-i-keep-records
https://www.irs.gov/businesses/small-businesses-self-employed/what-kind-of-records-should-i-keep
https://www.irs.gov/businesses/small-businesses-self-employed/how-long-should-i-keep-records
https://www.irs.gov/businesses/small-businesses-self-employed/how-should-i-record-my-business-transactions
https://www.irs.gov/businesses/small-businesses-self-employed/burden-of-proof
https://www.irs.gov/businesses/small-businesses-self-employed/employment-tax-recordkeeping

Tax years:
https://www.irs.gov/businesses/small-businesses-self-employed/tax-years

Instruction for 1128
https://www.irs.gov/pub/irs-dft/i1128--dft.pdf

Form 1128: Application To Adopt, Change, or Retain a Tax Year
https://www.irs.gov/pub/irs-pdf/f1128.pdf

Form I-9
https://www.uscis.gov/sites/default/files/document/forms/i-9.pdf

Form W-4
https://www.irs.gov/pub/irs-pdf/fw4.pdf

Employer's Tax guide
https://www.irs.gov/pub/irs-pdf/p15.pdf

Schedule K-2 (Form 1120-S)
https://www.irs.gov/pub/irs-pdf/f1120ssk.pdf

Instruction for K-2
https://www.irs.gov/pub/irs-pdf/i1120.pdf

Form 1040: U.S. Individual Income Tax Return
https://www.irs.gov/pub/irs-pdf/f1040.pdf

Instruction for 1040
https://www.irs.gov/pub/irs-pdf/i1040gi.pdf

Schedule C (Form 1040): Profit or Loss From Business
https://www.irs.gov/pub/irs-pdf/f1040sc.pdf

Instruction for Schedule C
https://www.irs.gov/pub/irs-pdf/i1040sc.pdf

Form 1040-SR: U.S. Income Tax Return for Seniors
https://www.irs.gov/pub/irs-pdf/f1040s.pdf

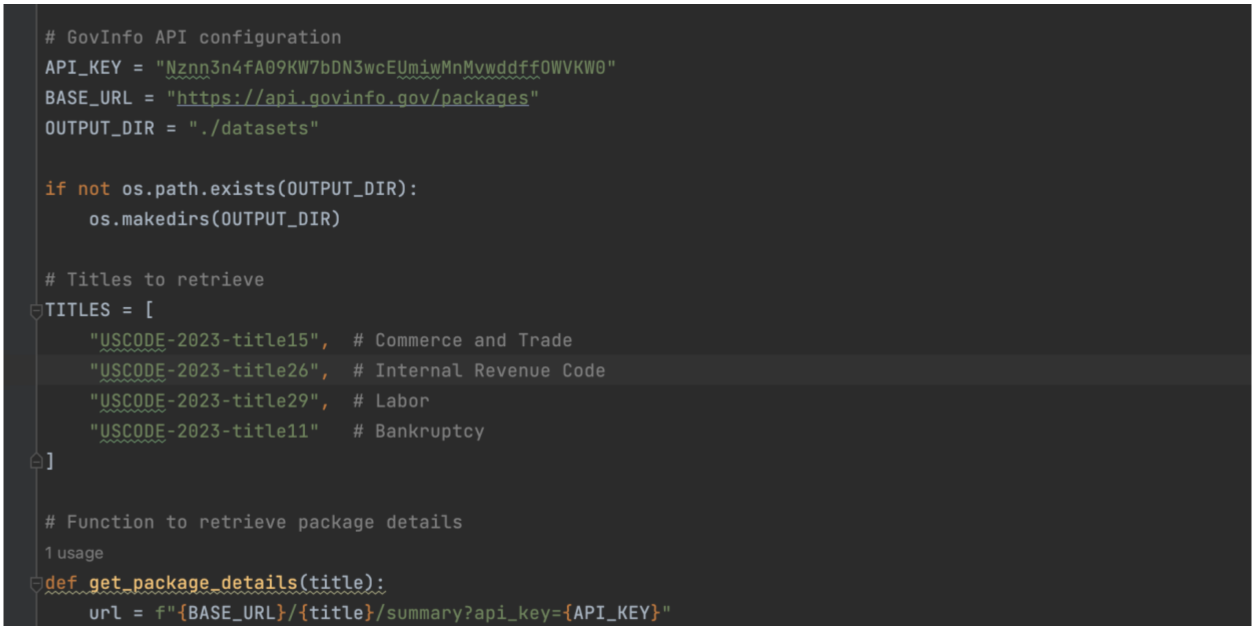
Form 1065: U.S. Return of Partnership Income
https://www.irs.gov/pub/irs-pdf/f1065.pdf

Potential Improvement:
To make our chatbot more realistic and ensure its output quality, we may need to feed some conversion scripts between clients and lawyers to let the chatbot learn the informal tone to make it sound more like a human instead of a robot. It could be challenging because most documents are written formally, and the chatbot would likely learn its pattern and output the same.

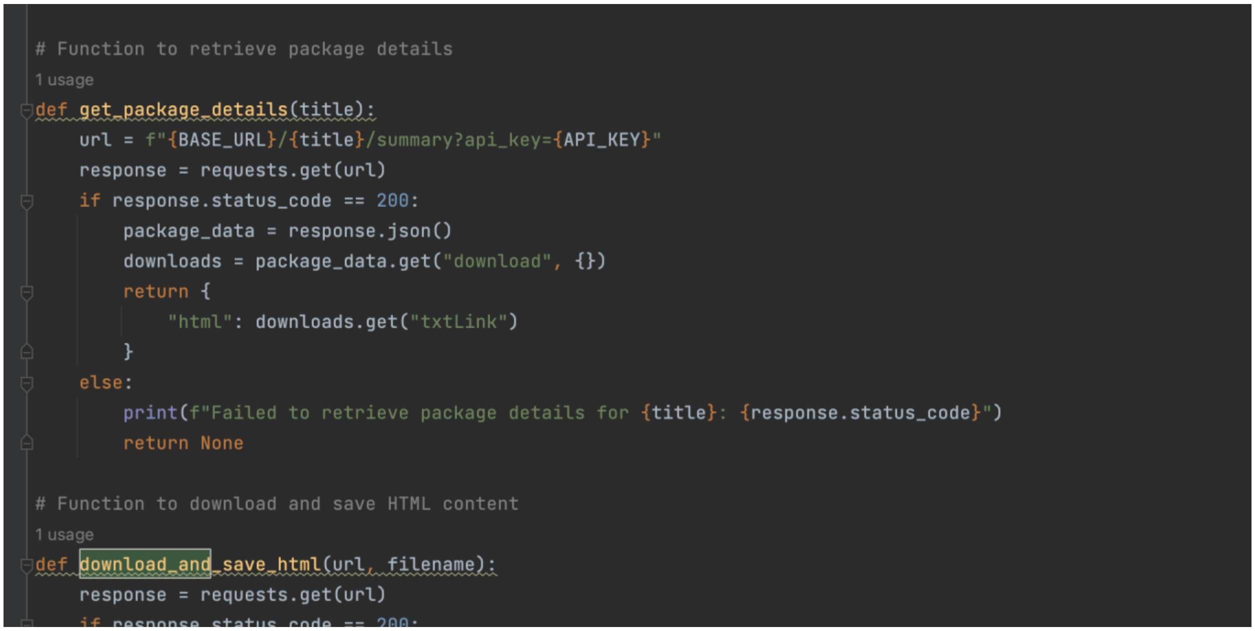
Data Exploration：

I used govinfo website to get the u.s.code context for our potential training dataset. https://api.govinfo.gov/docs/

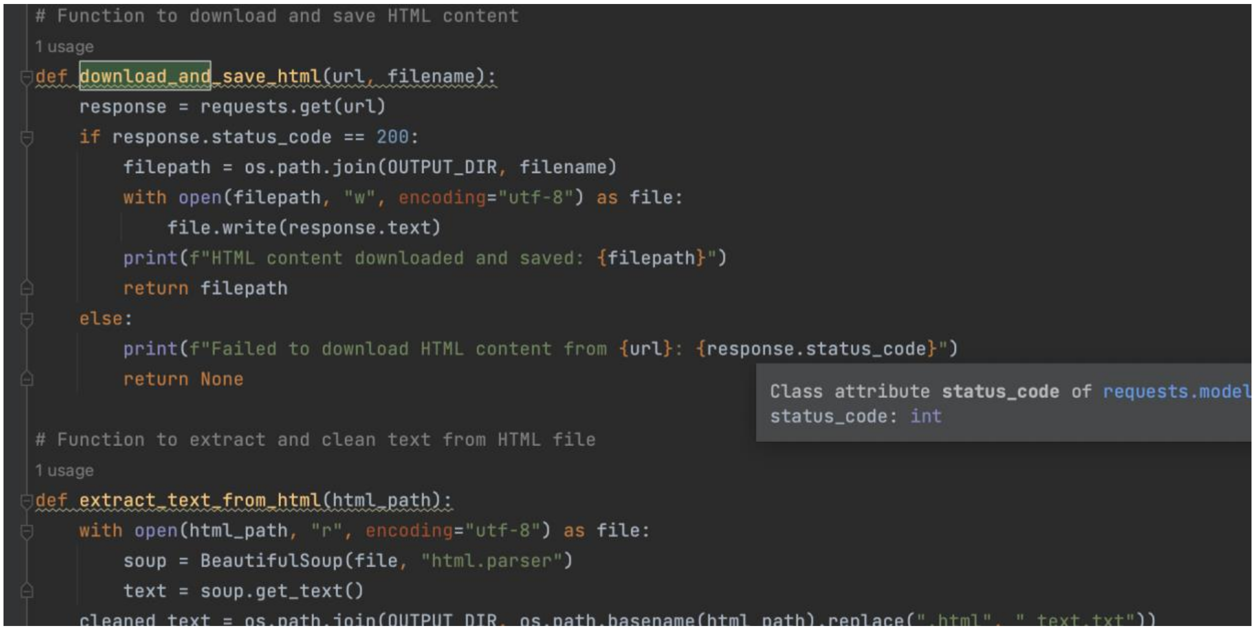
code:



```
# GovInfo API configuration
API_KEY = "Nznn3n4fA09KW7bDN3wcEUmiwMnMvwddff0WVKW0"
BASE_URL = "https://api.govinfo.gov/packages"
OUTPUT_DIR = "./datasets"

if not os.path.exists(OUTPUT_DIR):
    os.makedirs(OUTPUT_DIR)

# Titles to retrieve
TITLES = [
    "USCODE-2023-title15",  # Commerce and Trade
    "USCODE-2023-title26",  # Internal Revenue Code
    "USCODE-2023-title29",  # Labor
    "USCODE-2023-title11"   # Bankruptcy
]

# Function to retrieve package details
1 usage
def get_package_details(title):
    url = f"{BASE_URL}/{title}/summary?api_key={API_KEY}"
```

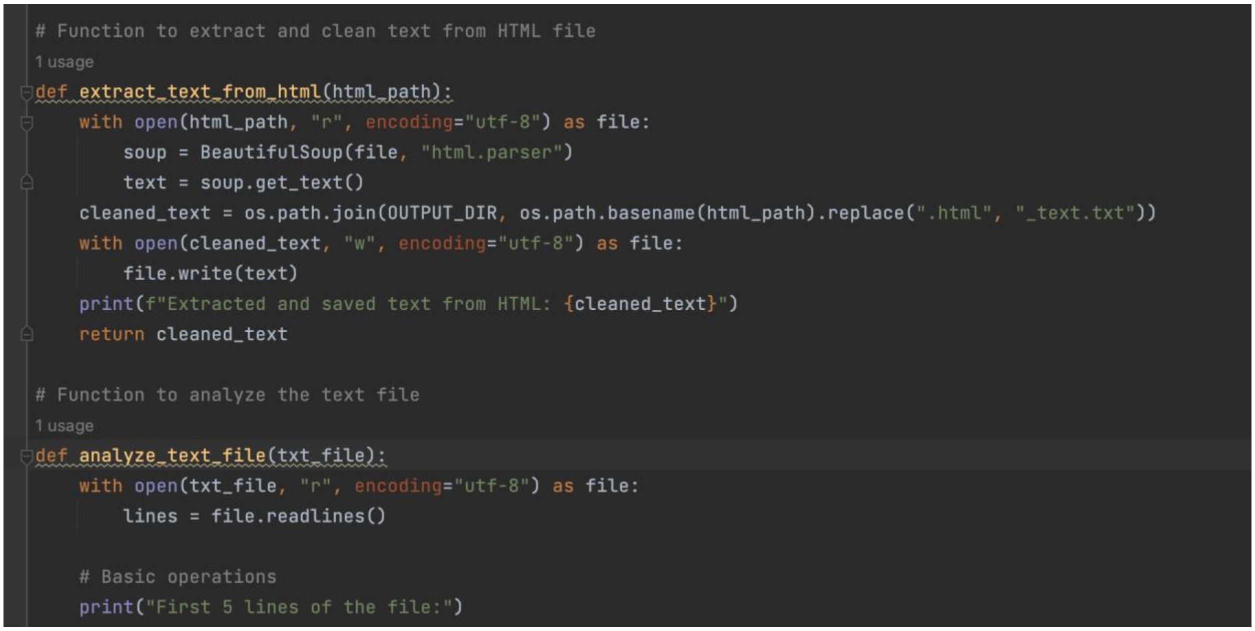


```
# Function to retrieve package details
1 usage
def get_package_details(title):
    url = f"{BASE_URL}/{title}/summary?api_key={API_KEY}"
    response = requests.get(url)
    if response.status_code == 200:
        package_data = response.json()
        downloads = package_data.get("download", {})
        return {
            "html": downloads.get("txtLink")
        }
    else:
        print(f"Failed to retrieve package details for {title}: {response.status_code}")
        return None

# Function to download and save HTML content
1 usage
def download_and_save_html(url, filename):
    response = requests.get(url)
    if response.status_code == 200:
```

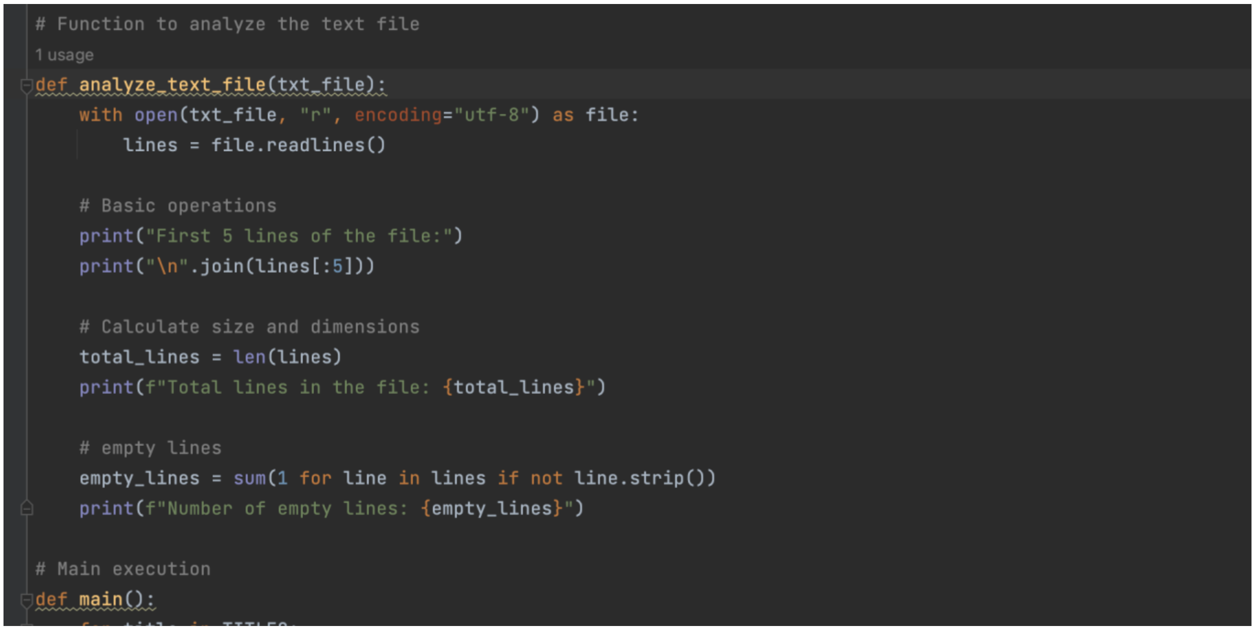


```
# Function to download and save HTML content
1 usage
def download_and_save_html(url, filename):
    response = requests.get(url)
    if response.status_code == 200:
        filepath = os.path.join(OUTPUT_DIR, filename)
        with open(filepath, "w", encoding="utf-8") as file:
            file.write(response.text)
        print(f"HTML content downloaded and saved: {filepath}")
        return filepath
    else:
        print(f"Failed to download HTML content from {url}: {response.status_code}")
        return None

# Function to extract and clean text from HTML file
1 usage
def extract_text_from_html(html_path):
    with open(html_path, "r", encoding="utf-8") as file:
        soup = BeautifulSoup(file, "html.parser")
        text = soup.get_text()
    cleaned_text = os.path.join(OUTPUT_DIR, os.path.basename(html_path).replace(".html", "_text.txt"))
```

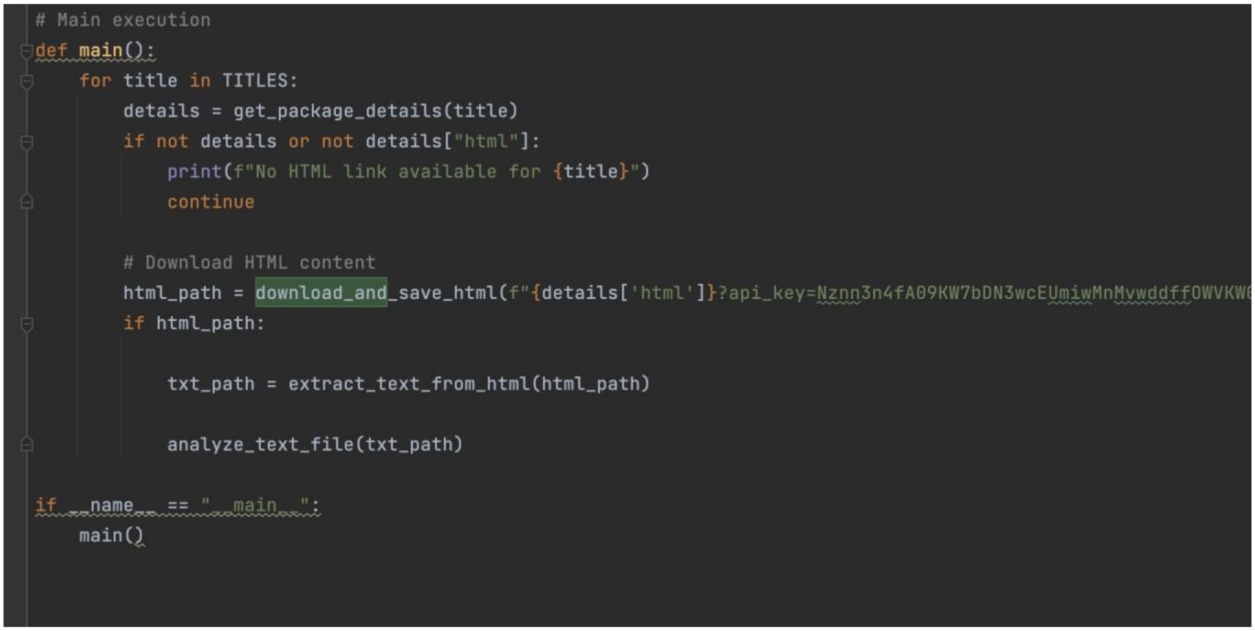
```
Class attribute status_code of requests.model
status_code: int
```



```
# Function to extract and clean text from HTML file
1 usage
def extract_text_from_html(html_path):
    with open(html_path, "r", encoding="utf-8") as file:
        soup = BeautifulSoup(file, "html.parser")
        text = soup.get_text()
    cleaned_text = os.path.join(OUTPUT_DIR, os.path.basename(html_path).replace(".html", "_text.txt"))
    with open(cleaned_text, "w", encoding="utf-8") as file:
        file.write(text)
    print(f"Extracted and saved text from HTML: {cleaned_text}")
    return cleaned_text

# Function to analyze the text file
1 usage
def analyze_text_file(txt_file):
    with open(txt_file, "r", encoding="utf-8") as file:
        lines = file.readlines()

    # Basic operations
    print("First 5 lines of the file:")
```

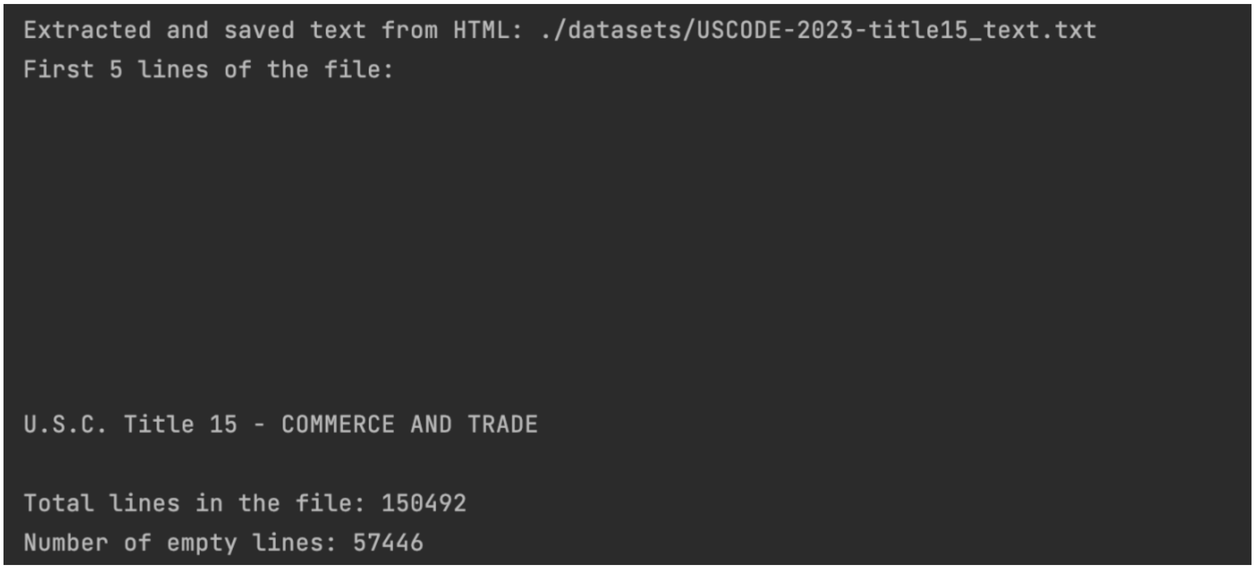


```python
# Function to analyze the text file
1 usage
def analyze_text_file(txt_file):
    with open(txt_file, "r", encoding="utf-8") as file:
        lines = file.readlines()

    # Basic operations
    print("First 5 lines of the file:")
    print("\n".join(lines[:5]))

    # Calculate size and dimensions
    total_lines = len(lines)
    print(f"Total lines in the file: {total_lines}")

    # empty lines
    empty_lines = sum(1 for line in lines if not line.strip())
    print(f"Number of empty lines: {empty_lines}")

# Main execution
def main():
```

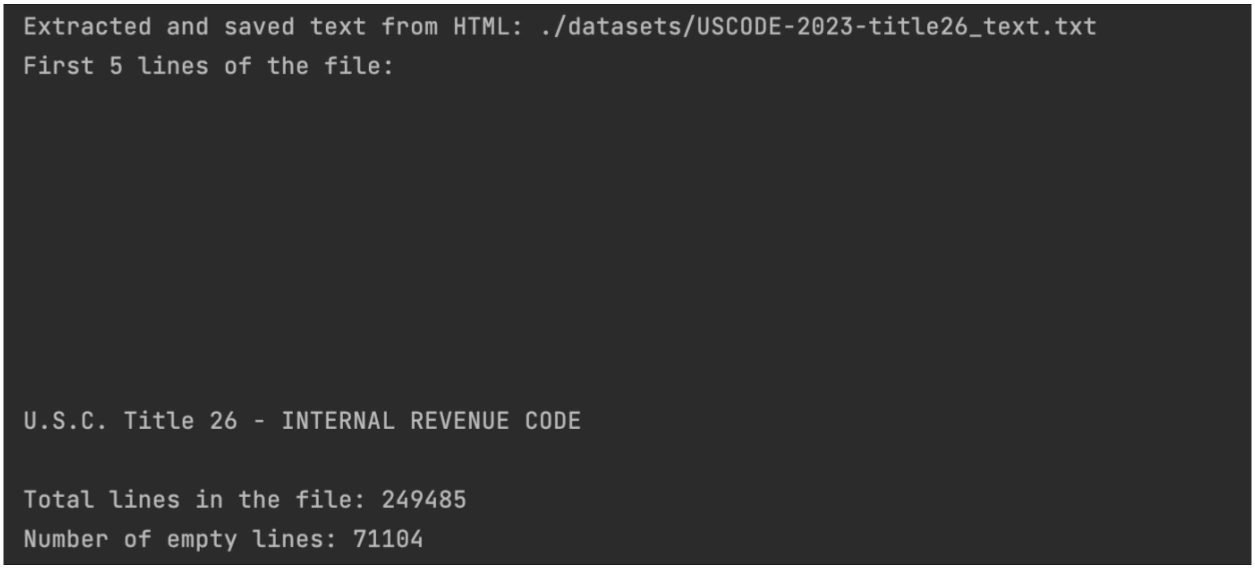


```python
# Main execution
def main():
    for title in TITLES:
        details = get_package_details(title)
        if not details or not details["html"]:
            print(f"No HTML link available for {title}")
            continue

        # Download HTML content
        html_path = download_and_save_html(f"{details['html']}?api_key=Nznn3n4fA09KW7bDN3wcEUmiwMnMvwddffOWVKW
        if html_path:

            txt_path = extract_text_from_html(html_path)

            analyze_text_file(txt_path)

if __name__ == "__main__":
    main()
```

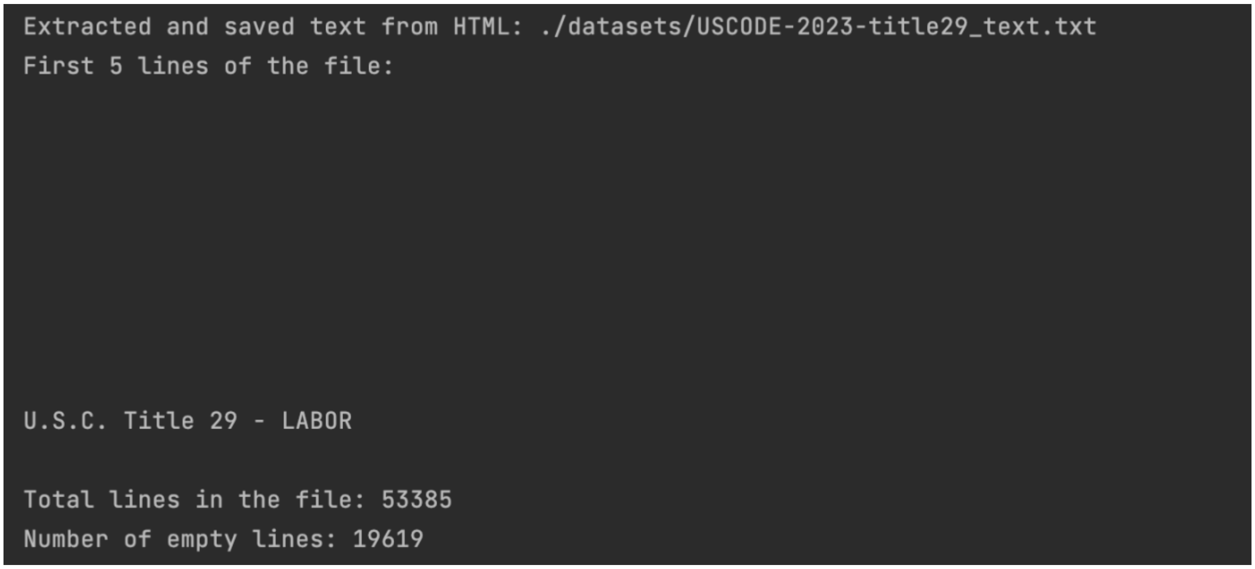
Analysis Output:



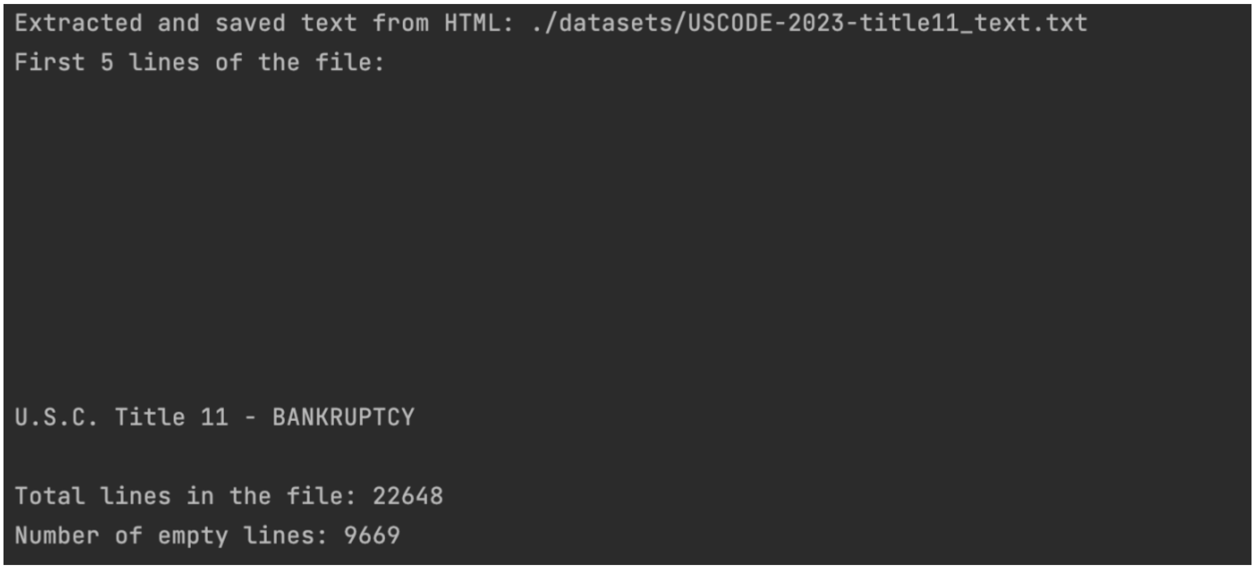
```
Extracted and saved text from HTML: ./datasets/USCODE-2023-title15_text.txt
First 5 lines of the file:








U.S.C. Title 15 - COMMERCE AND TRADE

Total lines in the file: 150492
Number of empty lines: 57446
```



```
Extracted and saved text from HTML: ./datasets/USCODE-2023-title26_text.txt
First 5 lines of the file:








U.S.C. Title 26 - INTERNAL REVENUE CODE

Total lines in the file: 249485
Number of empty lines: 71104
```



```
Extracted and saved text from HTML: ./datasets/USCODE-2023-title29_text.txt
First 5 lines of the file:








U.S.C. Title 29 - LABOR

Total lines in the file: 53385
Number of empty lines: 19619
```



```
Extracted and saved text from HTML: ./datasets/USCODE-2023-title11_text.txt
First 5 lines of the file:








U.S.C. Title 11 - BANKRUPTCY

Total lines in the file: 22648
Number of empty lines: 9669
```