```python
# Appendix C4 - content_importer.py

from .db_handler import DBHandler
from .tables import Document
from bs4 import BeautifulSoup
import requests
import time

# Future implementation: Tuning features by adding Documents' to their content. Maybe with a multiplier.
class ContentImporter(object):
    def __init__(self, db_name="klassify"):
        self.DBH = DBHandler(db_name, echo=False)
        self.ROOT_URL = "https://www.gov.uk"
        self.NON_RELEVANT_PHRASES = [
            "Skip to main content",
            "Find out more about cookies"
            "GOV.UK uses cookies to make the site simpler",
            "Is there anything wrong with this page",
            "Last updated",
            "Other ways to apply",
            "Before you start",
            "Elsewhere on the web",
            "Find out about call charges",
            "find out more about beta services",
            "Return to top ↑",
            "Find out more about cookies",
            "GOV.UK",
            "Don’t include personal or financial information",
            "Help us improve",
            "This file may not be suitable for users of assistive technology"
            "If you use assistive technology and need a version of this document in a more accessible format",
            "tell us what format you need It will help us if you say what assistive technology you use",
            "Request a different format",
            "What you were doing",
            "What went wrong",
            "uses cookies to make the site simpler."
        ]

    def parse_page(self, page):
        soup = BeautifulSoup(page, 'html.parser')
        return soup

    def extract_page_content(self, page):
        return page.text

    # Iterate through each Document in the database, get their URL on the site and
    # query it to obtain their HTML and eventually store it.
    def import_documents_html(self):
        documents = self.DBH.session.query(Document).all()

        count = 0
        for doc in documents:
            if doc.html == None:
                time.sleep(0.75)
                doc.html = requests.get(doc.web_url).text
                self.DBH.session.commit()
            count += 1
            if count % 250 == 0: print("Documents processed: %d/%d" %(count, len(documents)))

    # Iterate through the Documents' HTML, parse it and store it.
    def extract_documents_content(self):
        documents = self.DBH.session.query(Document).all()

```

```
64          count = 0
65          for doc in documents:
66              doc.content = self.extract_content(doc)
67              self.DBH.session.commit()
68              count += 1
69              if count % 250 == 0: print("Documents processed: %d/%d" %(count, len(document
s)))
70
71      def extract_content(self, document):
72          page = self.parse_page(document.html)
73          page = self.remove_unwanted_tags(page)
74          page = self.get_body(page)
75
76          page_content = self.extract_page_content(page)
77          page_content = self.remove_non_relevant_content(page_content)
78          page_content = self.remove_punctuaction_and_numbers(page_content)
79          return page_content
80
81      def get_body(self, page):
82          return page.body
83
84      # Discard anything inside footer, header and scripts
85      def remove_unwanted_tags(self, page):
86          for tag in page.find_all(['footer', 'script', 'header']):
87              tag.replace_with('')
88
89          return page
90
91      def remove_non_relevant_content(self, page):
92          for phrase in self.NON_RELEVANT_PHRASES:
93              page = page.replace(phrase, "")
94          return page
95
96      def remove_punctuaction_and_numbers(self, page):
97          punctuation = [ '\\', '>', '_', '`', '{', ']', '*', '[',
98                          '^', '+', '!', '(', ':', ';', "'", "’",
99                          '<', '|', '"', '?', '=', '}', '&', '/',
100                          '$', ')', '~', '#', '%', ',' ]
101          page = ''.join(ch for ch in page if ch not in punctuation)
102          page = ''.join([i for i in page if not i.isdigit()])
103          return page
104
```