Fundamentals of Programming-II

Project Report

2nd Semester



Instructor: Dr Saqib Nazeer/Affan Ahmed

Session: ME-15 Section: C Group:

SUBMITTED BY

Name	CMS ID
Haseeb Tahir	453901
Hasnain Ali	478806
Salar Azam	479001
Sakhawat Ali	470935

School of Mechanical & Manufacturing Engineering SMME

Code

```
import feedparser
import string
import time
import threading
from tkinter import *
from datetime import datetime
import html
import base64
# -----
# =========
# Code for retrieving and parsing
# Google and Yahoo News feeds
# ===========
def translate_html(text):
  111111
  Translates HTML entities to their corresponding characters.
  Also handles decoding if text is base64 encoded.
  111111
  # First, decode HTML entities
  text = html.unescape(text)
  # Check if the text is base64 encoded and decode it if necessary
  try:
   decoded_bytes = base64.decodebytes(text.encode('utf-8'))
    text = decoded_bytes.decode('utf-8')
```

```
except (base64.binascii.Error, UnicodeDecodeError):
    # If it's not base64 encoded, just return the HTML unescaped text
    pass
  return text
def process(url):
  Fetches news items from the rss url and parses them.
  Returns a list of NewsStory instances.
  feed = feedparser.parse(url)
  entries = feed.entries
  ret = []
  for entry in entries:
    guid = entry.guid
    title = translate_html(entry.title)
    link = entry.link
    # Check if description field exists
    if 'description' in entry:
      description = translate_html(entry.description)
    else:
       description = ""
    # Handling different date formats
    if 'published' in entry:
       pubdate_str = entry.published
    elif 'published_parsed' in entry:
```

```
pubdate_str = time.strftime('%a, %d %b %Y %H:%M:%S %Z', entry.published_parsed)
    else:
      continue
    try:
      pubdate = datetime.strptime(pubdate_str, "%a, %d %b %Y %H:%M:%S %Z")
    except ValueError:
      pubdate = datetime.strptime(pubdate_str, "%Y-%m-%dT%H:%M:%SZ")
    newsStory = NewsStory(guid, title, description, link, pubdate)
    ret.append(newsStory)
  return ret
# =========
# Data structure design
# ==========
class NewsStory:
  def __init__(self, guid, title, description, link, pubdate):
    self.guid = guid
    self.title = title
    self.description = description
    self.link = link
    self.pubdate = pubdate
  def get_guid(self):
    return self.guid
  def get_title(self):
```

```
def get_description(self):
    return self.description
  def get_link(self):
    return self.link
  def get_pubdate(self):
    return self.pubdate
# =========
# Triggers
# =========
class Trigger(object):
  def evaluate(self, story):
    raise NotImplementedError
class PhraseTrigger(Trigger):
  def __init__(self, phrase):
    self.phrase = phrase.lower()
  def is_phrase_in(self, text):
    text = text.lower()
    for char in string.punctuation:
      text = text.replace(char, ' ')
```

text_words = text.split()

return self.title

```
phrase_words = self.phrase.split()
    for i in range(len(text_words) - len(phrase_words) + 1):
       if text_words[i:i + len(phrase_words)] == phrase_words:
         return True
    return False
class TitleTrigger(PhraseTrigger):
  def evaluate(self, story):
    return self.is_phrase_in(story.get_title())
class DescriptionTrigger(PhraseTrigger):
  def evaluate(self, story):
    return self.is_phrase_in(story.get_description())
class TimeTrigger(Trigger):
  def __init__(self, time):
    self.time = datetime.strptime(time, "%Y-%m-%dT%H:%M:%SZ")
class BeforeTrigger(TimeTrigger):
  def evaluate(self, story):
    return story.get_pubdate() < self.time
class AfterTrigger(TimeTrigger):
  def evaluate(self, story):
    return story.get_pubdate() > self.time
```

```
class NotTrigger(Trigger):
  def __init__(self, trigger):
    self.trigger = trigger
  def evaluate(self, story):
    return not self.trigger.evaluate(story)
class AndTrigger(Trigger):
  def __init__(self, trigger1, trigger2):
    self.trigger1 = trigger1
    self.trigger2 = trigger2
  def evaluate(self, story):
    return self.trigger1.evaluate(story) and self.trigger2.evaluate(story)
class OrTrigger(Trigger):
  def __init__(self, trigger1, trigger2):
    self.trigger1 = trigger1
    self.trigger2 = trigger2
  def evaluate(self, story):
    return self.trigger1.evaluate(story) or self.trigger2.evaluate(story)
# ==========
# Filtering
```

```
# =========
def filter_stories(stories, triggerlist):
  filtered_stories = []
  for story in stories:
    for trigger in triggerlist:
      if trigger.evaluate(story):
        filtered_stories.append(story)
        break
  return filtered_stories
# =========
# User-Specified Triggers
# =========
def read_trigger_config(filename):
  trigger_file = open(filename, 'r')
  lines = []
  for line in trigger_file:
    line = line.rstrip()
    if not (len(line) == 0 or line.startswith('//')):
      lines.append(line)
  trigger_file.close()
  triggers = {}
  trigger_list = []
  for line in lines:
    parts = line.split(',')
```

```
if parts[0] == 'ADD':
      for name in parts[1:]:
         if name in triggers:
           trigger_list.append(triggers[name])
    else:
      trigger_name = parts[0]
      trigger_type = parts[1]
      if trigger_type == 'TITLE':
         triggers[trigger_name] = TitleTrigger(parts[2])
      elif trigger_type == 'DESCRIPTION':
         triggers[trigger_name] = DescriptionTrigger(parts[2])
      elif trigger_type == 'AFTER':
        triggers[trigger_name] = AfterTrigger(parts[2])
      elif trigger_type == 'BEFORE':
         triggers[trigger_name] = BeforeTrigger(parts[2])
      elif trigger_type == 'NOT':
         if parts[2] in triggers:
           triggers[trigger_name] = NotTrigger(triggers[parts[2]])
      elif trigger_type == 'AND':
         if parts[2] in triggers and parts[3] in triggers:
           triggers[trigger_name] = AndTrigger(triggers[parts[2]], triggers[parts[3]])
      elif trigger_type == 'OR':
         if parts[2] in triggers and parts[3] in triggers:
           triggers[trigger_name] = OrTrigger(triggers[parts[2]], triggers[parts[3]])
  return trigger_list
# =========
# Main Thread
```

```
# ===========
SLEEPTIME = 120 # seconds
def main_thread(master, keywords):
  try:
    triggerlist = []
    if keywords:
      for keyword in keywords:
        triggerlist.append(OrTrigger(TitleTrigger(keyword)), DescriptionTrigger(keyword)))
    frame = Frame(master)
    frame.pack(side=BOTTOM)
    scrollbar = Scrollbar(master)
    scrollbar.pack(side=RIGHT, fill=Y)
    t = "Google & Yahoo Top News"
    title = StringVar()
    title.set(t)
    ttl = Label(master, textvariable=title, font=("Helvetica", 18))
    ttl.pack(side=TOP)
    cont = Text(master, font=("Helvetica", 14), yscrollcommand=scrollbar.set)
    cont.pack(side=BOTTOM)
    cont.tag_config("title", justify='center')
    button = Button(frame, text="Exit", command=master.destroy)
    button.pack(side=BOTTOM)
    guidShown = []
    def get_cont(newstory):
```

```
if newstory.get_guid() not in guidShown:
       cont.insert(END, newstory.get_title() + "\n", "title")
       cont.insert(END, "\n----\n", "title")
       cont.insert(END, newstory.get_description())
       cont.insert(END,
guidShown.append(newstory.get_guid())
   while True:
     print("Polling...")
     stories = process("http://news.google.com/news?output=rss")
     stories.extend(process("http://news.yahoo.com/rss/topstories"))
     stories = filter_stories(stories, triggerlist)
     list(map(get_cont, stories))
     scrollbar.config(command=cont.yview)
     time.sleep(SLEEPTIME)
 except Exception as e:
   print(f"Error occurred: {e}")
if __name__ == '__main__':
 root = Tk()
 root.title("RSS Feed Filter")
 keywords = input("Enter keywords (comma-separated): ").strip().split(',')
 keywords = [keyword.strip() for keyword in keywords if keyword.strip()]
```

```
t = threading.Thread(target=main_thread, args=(root, keywords))
t.start()
```

Output

RSS Feed Filter

root.mainloop()

```
Python 3.11.5 | packaged by Anaconda, Inc. | (main, Sep 11 2023, 13:26:23) [MSC v.1916 64 bit (AMD64)] Type "copyright", "credits" or "license" for more information.

IPython 8.15.0 -- An enhanced Interactive Python.

In [1]: runfile('D:/ME-15/2nd Semester/Fundamentals Of Programming II/Lab Manuals/Project/project.py', wdir='D:/ME-15/2nd Semester/Fundamentals Of Programming II/Lab Manuals/Project')

Enter keywords (comma-separated): Pakistan,India,Iran,Imran Khan
Polling...
```

Google & Yahoo Top News

PM Shehbaz meets Ayatollah Khamenei after Tehran funeral for Raisi - DAWN.com

<a href="https://news.google.com/rss/articles/CBMiIWh0dHBzOi8vd3d3LmRhd24uY29tL25ld3M</p> /MTgzNTAwNtIBJWh0dHBzOi8vd3d3LmRhd24uY29tL25ld3MvYW1wLzE4MzUwMDY?oc=5" target=" >lank">PM Shehbaz meets Ayatollah Khamenei after Tehran funeral for Raisi DAWN.comDAWN.comDAWN.com ZS5jb20ucGsvc3RvcnkvMjQ2NzgwNi9wbS1zaGVoYmF6LWF0dGVuZHMtbWVtb3JpYWwtZm9yLXByZ KNpZGVudC1yYWlzaS1pbi10ZWhyYW7SAWRodHRwczovL3RyaWJ1bmUuY29tLnBrL3N0b3J5LzI0Nj :4MDYvcG0tc2hlaGJhei1hdHRlbmRzLW1lbW9yaWFsLWZvci1wcmVzaWRlbnQtcmFpc2ktaW4tdGVoc nFuP2FtcD0x?oc=5" target="_blank">Raisi's successor committed to strengthening Pakistan ties, Kha nenei assures Shehbaz The Express TribunePM Shehbaz meets Iran's supreme lea ler to condole President Raisi's tragic death The NationLeader calls for upgrading of elations with Pakistan اير نااه href="https://news.google.com/rs" 3/articles/CBMiV2h0dHBzOi8vd3d3LnJhZGlvLmdvdi5way8yMi0wNS0yMDI0L3BtLWxlYXZlcy1mb3ItaG9 ZWxhbmQtYWZ0ZXItY29tcGxldGluZy1pcmFucy12aXNpdNIBAA?oc=5" target="_blank">PM leaves for nomeland after completing Iran's visit Radio Pakistan