****

**COMP5324 Internet Information Retrieval Report**

**Assignment 2: A Web Spider**

|  |  |
| --- | --- |
| **Name** | **Student ID** |
| Chen Qi | 18083945G |
| Zhao Haiqi | 18087348G |

### 

TOC

[Readme 2](#_Toc6154942)

[Functions of the program 3](#_Toc6154943)

[Spider.py 3](#_Toc6154944)

[Introduction of the development 3](#_Toc6154945)

[Basic Requirement 3](#_Toc6154946)

[Advanced Requirements 4](#_Toc6154947)

[Superior Advanced Requirements 4](#_Toc6154948)

# Readme

For readme file, you can also check our Github’s Repo for **Installation Guide** and **User Manual** here.

[git@github.com:Cqyhid/Web-Crawler.git](mailto:git@github.com:Cqyhid/Web-Crawler.git)

# Functions of the program

## Spider.py

#### \_\_fetch\_content(self, url):

Get the content from html file.

#### \_\_analysis (self, htmls):

Analysis the content of html file ,get the hyperlink from the html file.

#### \_\_save\_html(self, file\_name, file\_content):

Save and download the html from internet to local storage.

#### go(self, search\_mode):

Include DFS and BFS. It keeping searching the content until the end.

#### check\_content(self, key, result\_dict):

#### Add indention for the displaying results.

#### result\_sorting(self, result):

#### Add indention for displaying results.

# 

# Introduction of the development

## Basic Requirement

First, to develop the spider program. We choose “List of television channels “ for our assignment. We use chrome browser to browse the wiki webpage. Using development mode, check the element of the website. The requirement is to get the content of subcategory, so we found all the <div> element from the website. We checkout all the class name from those and figure out the content of each elements. Then we found that there is a class name called “CategoryTreeItem”. So we design a regular expression to get the relevant class in our program. We use the package ‘requests’ in python to fetch the content from the website. And after we get the <div> element, we need to get the hyperlink from the content in the html file. We design another regular expression to get the hyperlink.

For BFS and DFS, we use a list to do that. For BFS, we pop the first element from the content, then add the others after it. It’s LIFO. For DFS, It’s FIFO, we pop the last element from the content.

## Advanced Requirements

The advanced requirements is to categorize the contents by relationship using indention or other method. We use recursion to calculate the number of indention of each subcategory’s name. We store all the data in dictionary. In the dictionary, the key of element is the name of website. The value is a list which contains all subcategory’s name.

## Superior Advanced Requirements

We save all the content in the directory locally. We need to encode the file into utf-8 and decode to cp950. Then we can use our system from assignment 1 to load the files. And the function are works well as Assignment 1.

A picture containing photo, indoor, computer

Description automatically generated