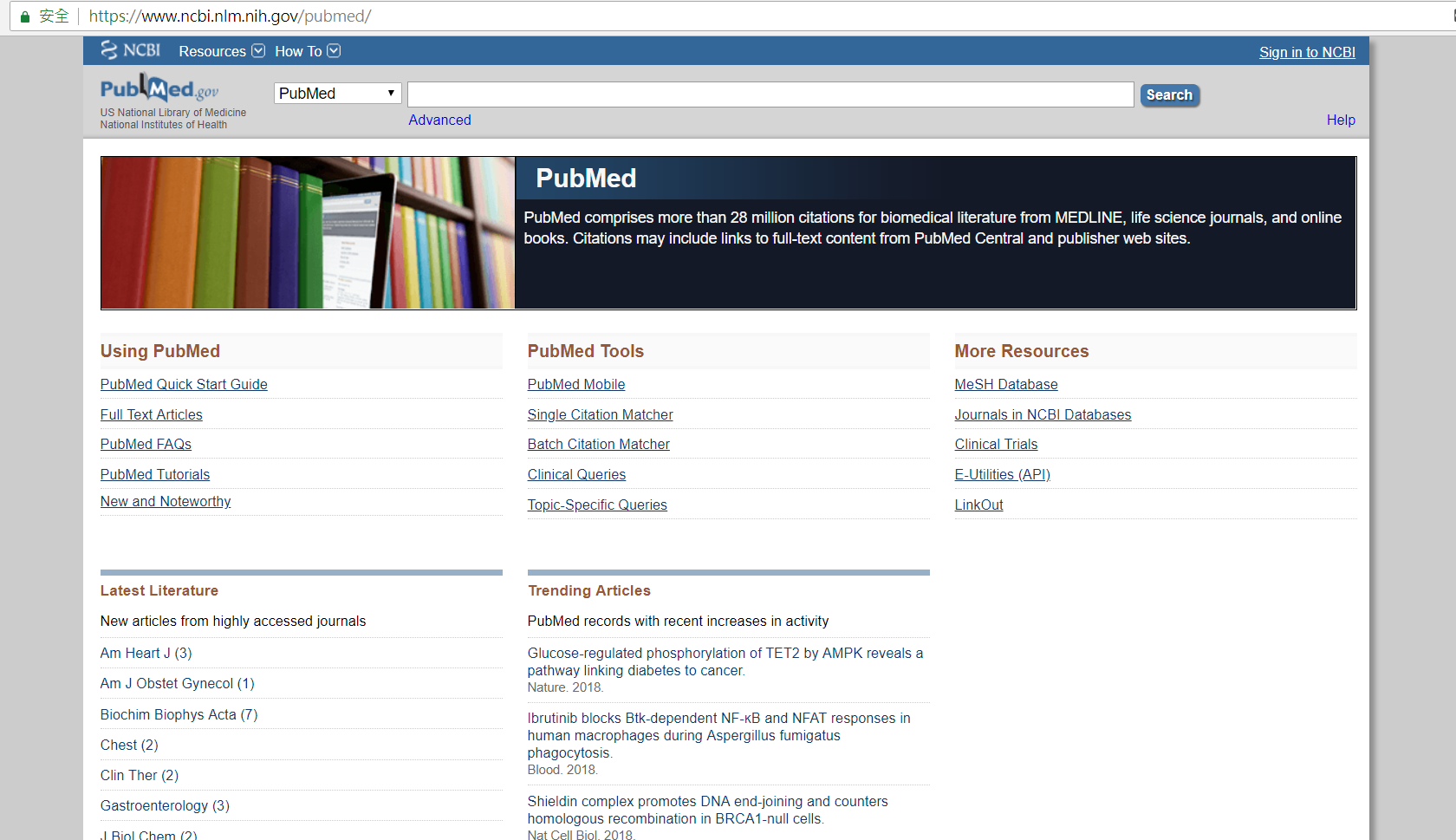
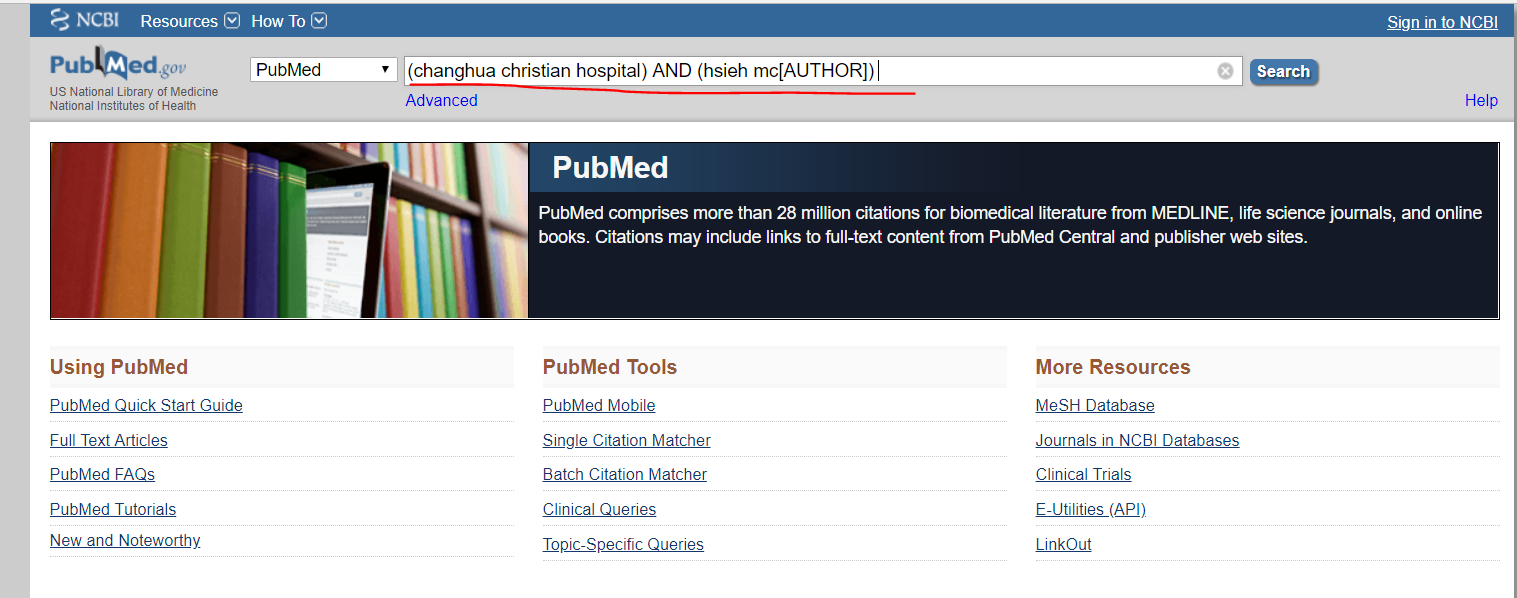
1. PURPOSE 
   1. If we want to go to special website (like PubMed website), like search Specific doctors and specific hospitals about their article. We can collect what we want the all information by manual before. Now, you can use web crawler skill to grab all information automatically.
2. DEFINITIONS/ABBREVIATIONS
   1. Web crawler: A Web crawler starts with a list of [URLs](https://en.wikipedia.org/wiki/Uniform_Resource_Locator) to visit, called the seeds. As the crawler visits these URLs, it identifies all the [hyperlinks](https://en.wikipedia.org/wiki/Hyperlink) in the page and adds them to the list of URLs to visit, called the [crawl frontier](https://en.wikipedia.org/wiki/Crawl_frontier). URLs from the frontier are [recursively](https://en.wikipedia.org/wiki/Recursion) visited according to a set of policies. If the crawler is performing archiving of [websites](https://en.wikipedia.org/wiki/Website) it copies and saves the information as it goes. The archives are usually stored in such a way they can be viewed, read and navigated as they were on the live web, but are preserved as ‘snapshots'
   2. R language: **R** is a [programming language](https://en.wikipedia.org/wiki/Programming_language) and [free](https://en.wikipedia.org/wiki/Free_software) software environment for [statistical computing](https://en.wikipedia.org/wiki/Statistical_computing) and graphics that is supported by the R Foundation for Statistical Computing
   3. XML: In [computing](https://en.wikipedia.org/wiki/Computing), Extensible Markup Language (XML) is a [markup language](https://en.wikipedia.org/wiki/Markup_language) that defines a set of rules for encoding [documents](https://en.wikipedia.org/wiki/Electronic_document) in a [format](https://en.wikipedia.org/wiki/File_format) that is both [human-readable](https://en.wikipedia.org/wiki/Human-readable_medium) and [machine-readable](https://en.wikipedia.org/wiki/Machine-readable_data)
3. PROCEDURE *[Only relevant for SOPs-Include a set of directions or description of steps to achieve the purpose stated above; this may include standard work and process flows. Where appropriate, include roles/responsibilities]*
   1. First, we want to go to PubMed to search all paper of Hsieh mc of Changhua Christian Hospital . we go to the PubMeb website=>

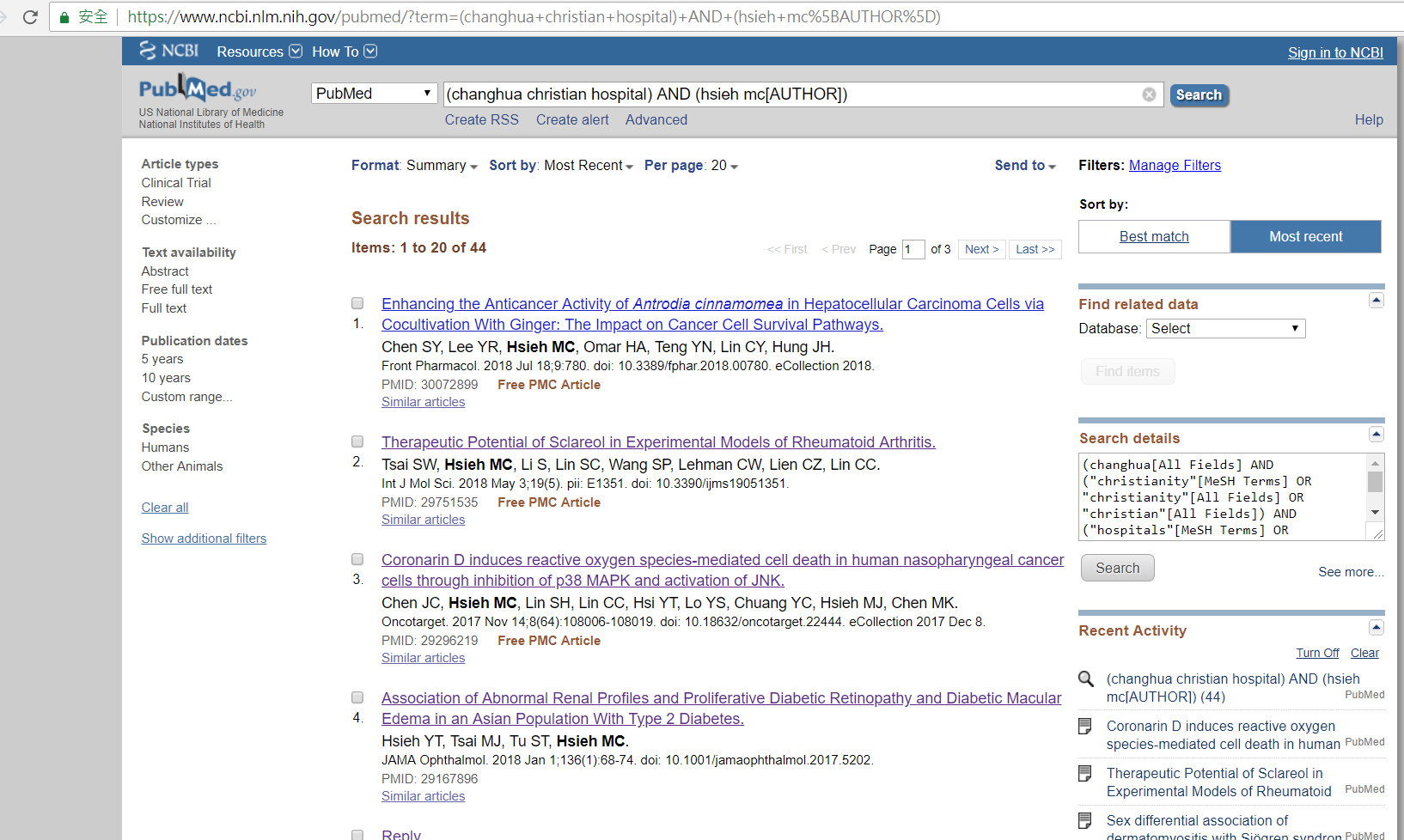


The URL is : <https://www.ncbi.nlm.nih.gov/pubmed/>

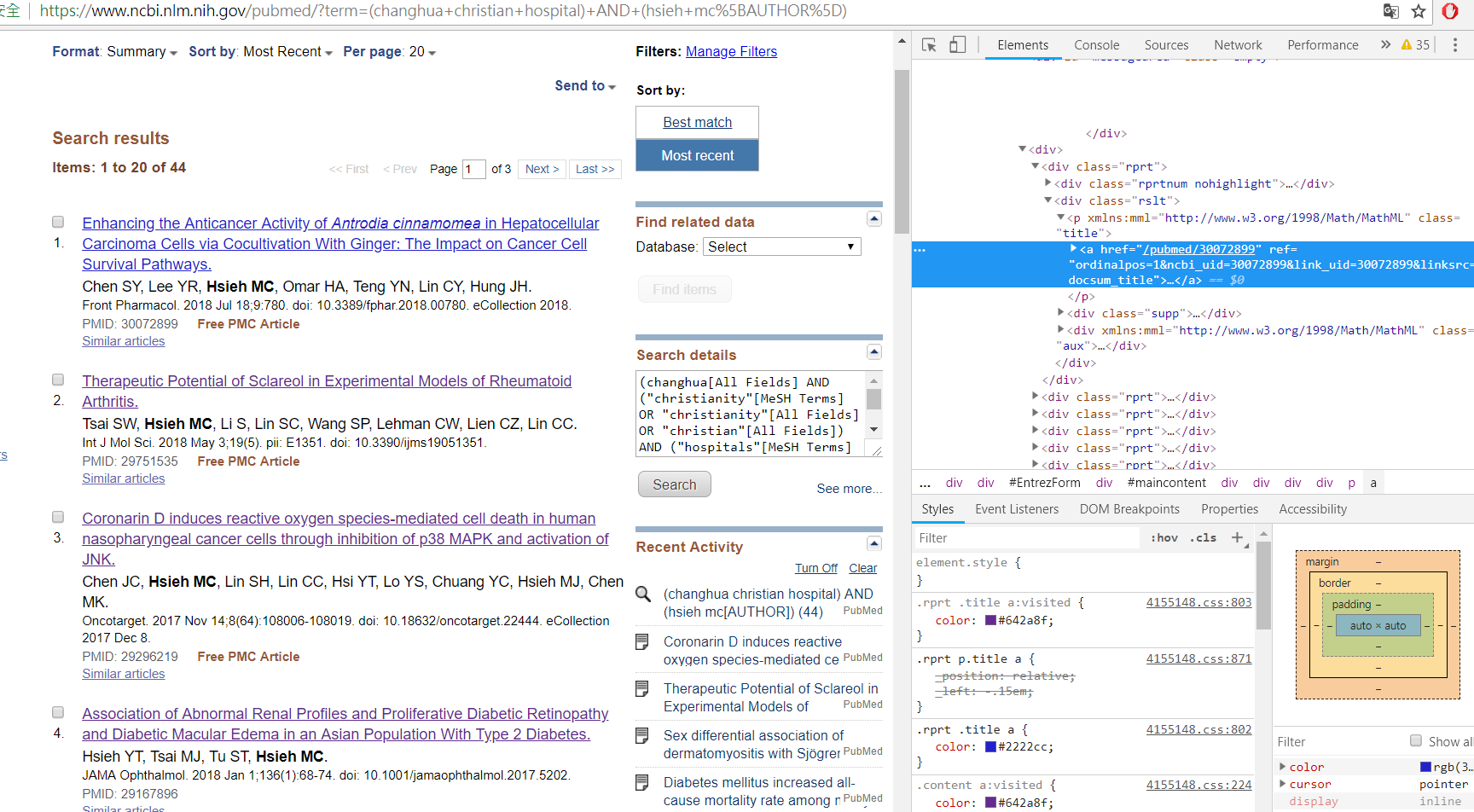
* 1. Second, we search what we want to search doctor and hospital and put to the website:



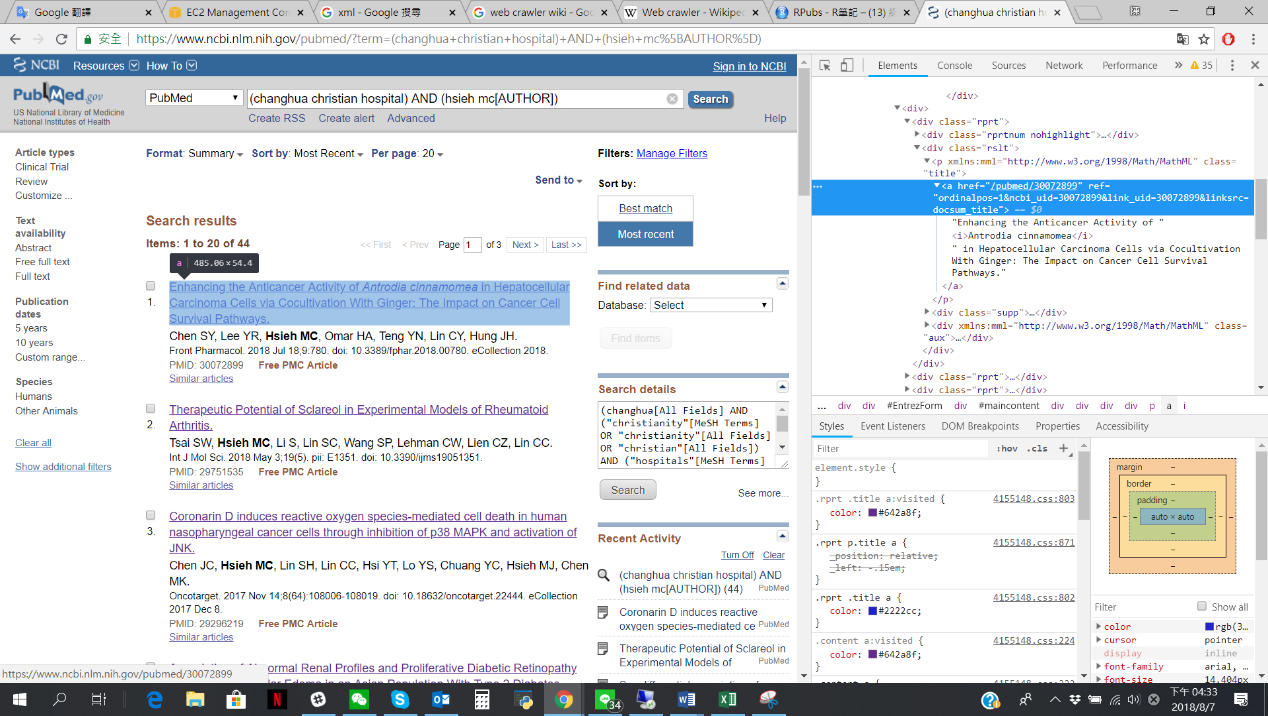
The URL is changed: <https://www.ncbi.nlm.nih.gov/pubmed/?term=(changhua+christian+hospital)+AND+(hsieh+mc%5BAUTHOR%5D)>



* 1. Third, we want to know their article title, title of journal, author, Year of publication for each paper(20 paper) => we need to install this packages: xml2



Decrypt the "title" of the file and the corresponding "hyperlink"

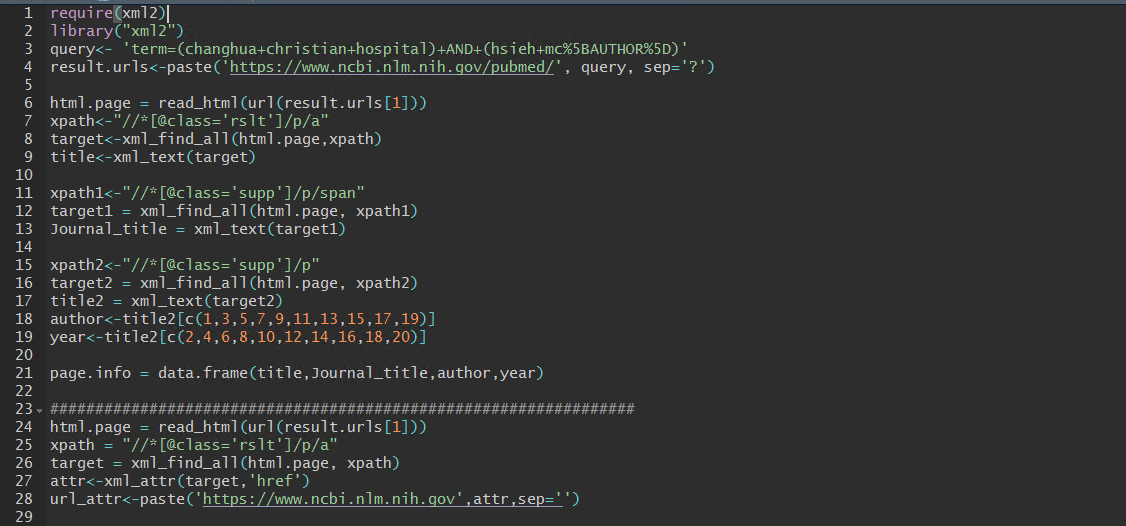


When you move the mouse to the title of the decrypted file, the following phenomenon occurs:



The corresponding html code will be displayed in the Detector:

The R code is under this :



Require(xml2): means call your package.

query<- 'term=(changhua+christian+hospital)+AND+(hsieh+mc%5BAUTHOR%5D)' : means find their search URL.

result.urls<-paste('https://www.ncbi.nlm.nih.gov/pubmed/', query, sep='?') : means paste office website URL with search URL.

html.page = read\_html(url(result.urls[1])): means read the url and save the html page as an object

xpath<-"//\*[@class='rslt']/p/a" :

target<-xml\_find\_all(html.page,xpath):

means write an xpath based on the information in the image above.

title<-xml\_text(target) : means Decrypt the "title" of the file.

Do the same things as the above, find the Journal\_title

xpath1<-"//\*[@class='supp']/p/span"

target1 = xml\_find\_all(html.page, xpath1)

Journal\_title = xml\_text(target1)

Do the same things as the above, find the author and year of publishing

xpath2<-"//\*[@class='supp']/p"

target2 = xml\_find\_all(html.page, xpath2)

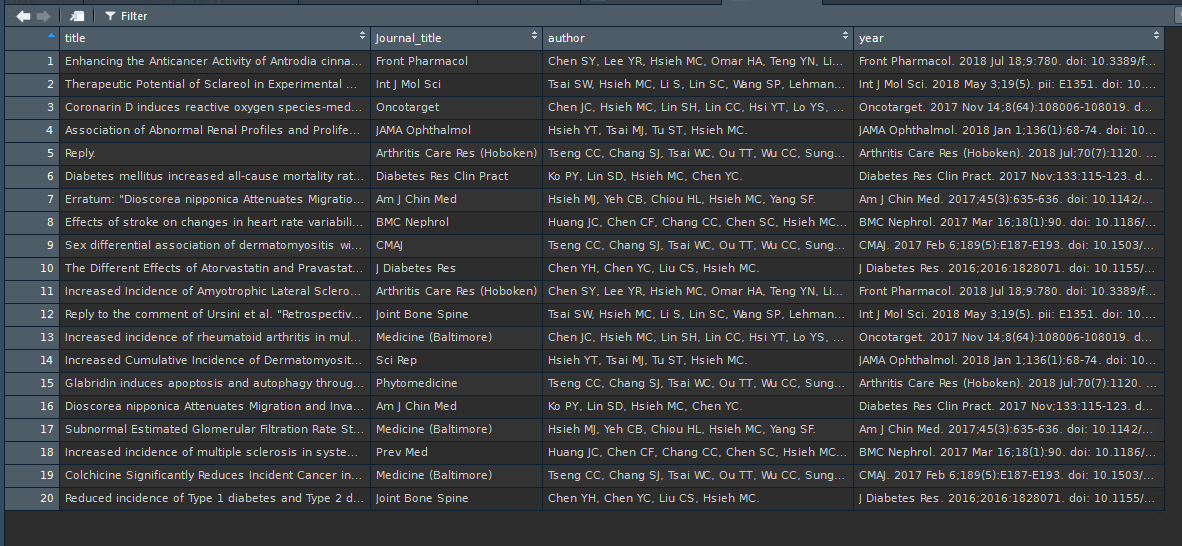
title2 = xml\_text(target2)

author<-title2[c(1,3,5,7,9,11,13,15,17,19)]

year<-title2[c(2,4,6,8,10,12,14,16,18,20)]

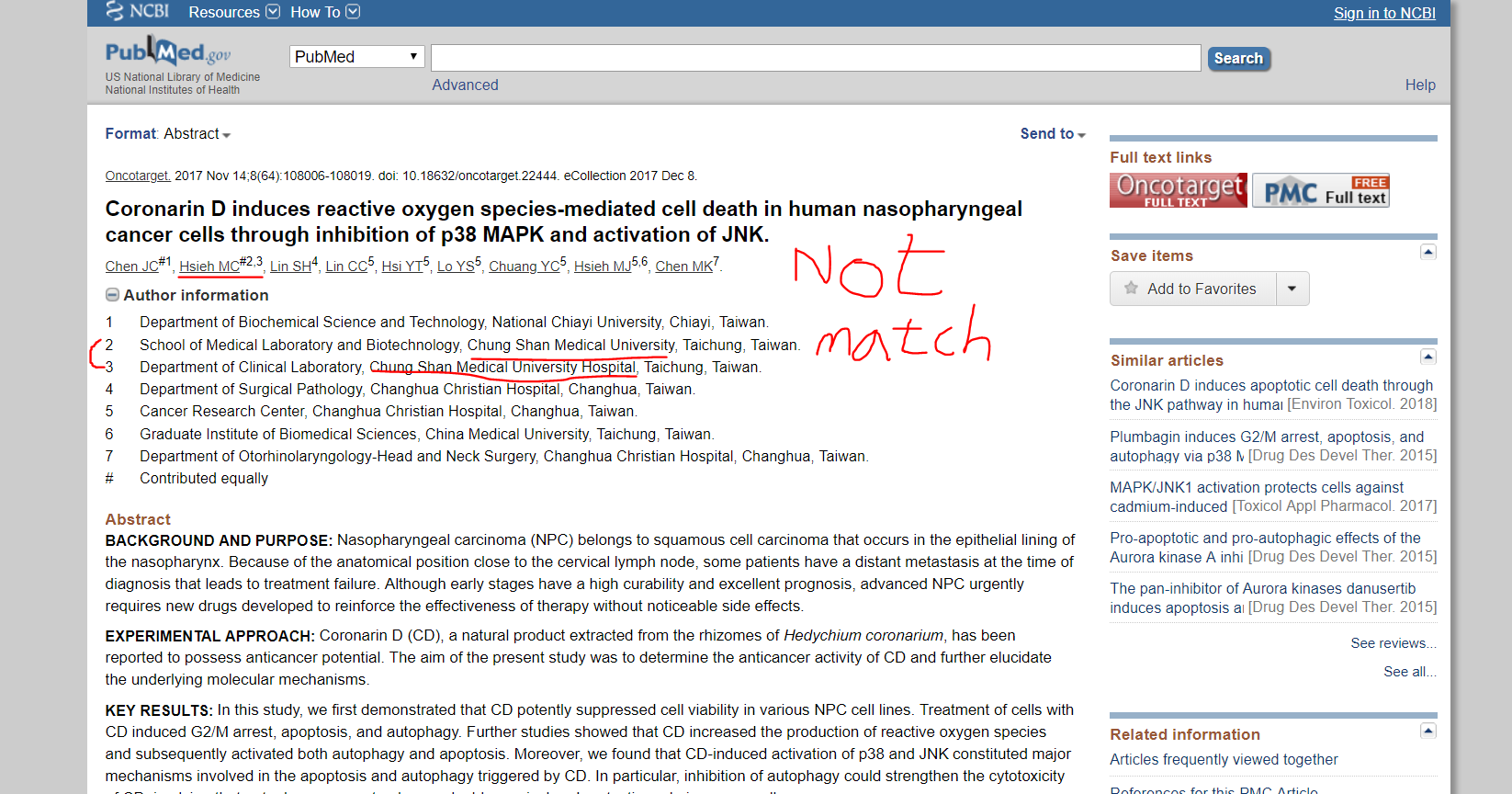
Now, make a dataframe including article title, title of journal, author, Year of publication for each paper(20 paper)

page.info = data.frame(title,Journal\_title,author,year)

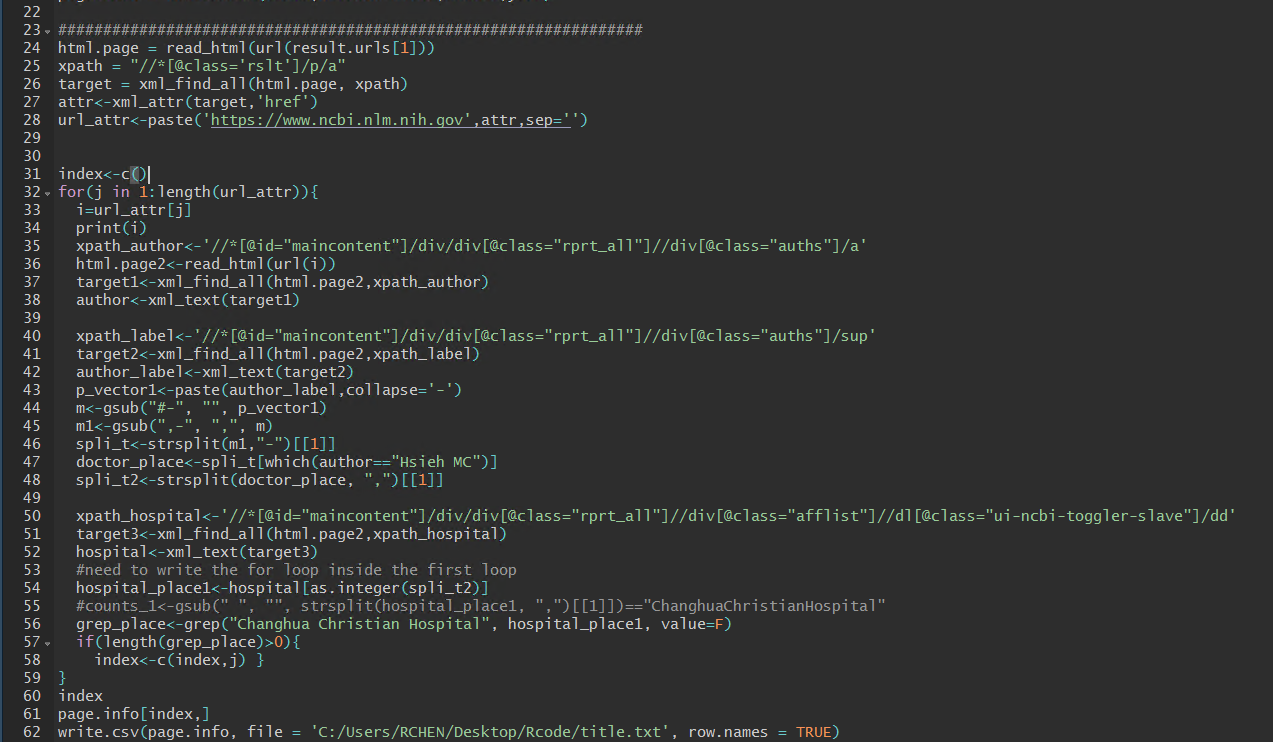


* 1. Finally, we can grab all the information in twenty paper.
  2. Actually, this have a bug: if we click third paper, the name of doctor doesn’t match

Hospital name. So we have to write a loop to judge whether the doctor match hospital.



The R code is under this :



* 1. Under this loop, they will output which paper should we leave:



* 1. Finally, output become this:



Now, we can go to website to check is correct or not.

1. REFERENCES (EXTERNAL) *[Include external references that are pertinent to the content of this document]*
   1. Example:<https://rpubs.com/skydome20/R-Note13-Web-Crawler-on-CIA-CREST-by-xml2>
   2. <https://en.wikipedia.org/wiki/XML>
   3. <https://en.wikipedia.org/wiki/R_(programming_language)>
   4. <https://en.wikipedia.org/wiki/Web_crawler>
2. RELATED DOCUMENTS *[Include list of attachments such as related forms, templates, or charts and records]*
3. KEY WORDS
   1. Crawlers
   2. R language
   3. XML

10. APPENDIX (Any websites text, troubleshooting)

Rcode:

require(xml2)

library("xml2")

query<- 'term=(changhua+christian+hospital)+AND+(hsieh+mc%5BAUTHOR%5D)'

result.urls<-paste('https://www.ncbi.nlm.nih.gov/pubmed/', query, sep='?')

html.page = read\_html(url(result.urls[1]))

xpath<-"//\*[@class='rslt']/p/a"

target<-xml\_find\_all(html.page,xpath)

title<-xml\_text(target)

xpath1<-"//\*[@class='supp']/p/span"

target1 = xml\_find\_all(html.page, xpath1)

Journal\_title = xml\_text(target1)

xpath2<-"//\*[@class='supp']/p"

target2 = xml\_find\_all(html.page, xpath2)

title2 = xml\_text(target2)

author<-title2[c(1,3,5,7,9,11,13,15,17,19)]

year<-title2[c(2,4,6,8,10,12,14,16,18,20)]

page.info = data.frame(title,Journal\_title,author,year)

#################################################################

html.page = read\_html(url(result.urls[1]))

xpath = "//\*[@class='rslt']/p/a"

target = xml\_find\_all(html.page, xpath)

attr<-xml\_attr(target,'href')

url\_attr<-paste('https://www.ncbi.nlm.nih.gov',attr,sep='')

index<-c()

for(j in 1:length(url\_attr)){

i=url\_attr[j]

print(i)

xpath\_author<-'//\*[@id="maincontent"]/div/div[@class="rprt\_all"]//div[@class="auths"]/a'

html.page2<-read\_html(url(i))

target1<-xml\_find\_all(html.page2,xpath\_author)

author<-xml\_text(target1)

xpath\_label<-'//\*[@id="maincontent"]/div/div[@class="rprt\_all"]//div[@class="auths"]/sup'

target2<-xml\_find\_all(html.page2,xpath\_label)

author\_label<-xml\_text(target2)

p\_vector1<-paste(author\_label,collapse='-')

m<-gsub("#-", "", p\_vector1)

m1<-gsub(",-", ",", m)

spli\_t<-strsplit(m1,"-")[[1]]

doctor\_place<-spli\_t[which(author=="Hsieh MC")]

spli\_t2<-strsplit(doctor\_place, ",")[[1]]

xpath\_hospital<-'//\*[@id="maincontent"]/div/div[@class="rprt\_all"]//div[@class="afflist"]//dl[@class="ui-ncbi-toggler-slave"]/dd'

target3<-xml\_find\_all(html.page2,xpath\_hospital)

hospital<-xml\_text(target3)

#need to write the for loop inside the first loop

hospital\_place1<-hospital[as.integer(spli\_t2)]

#counts\_1<-gsub(" ", "", strsplit(hospital\_place1, ",")[[1]])=="ChanghuaChristianHospital"

grep\_place<-grep("Changhua Christian Hospital", hospital\_place1, value=F)

if(length(grep\_place)>0){

index<-c(index,j) }

}

index

page.info[index,]

write.csv(page.info, file = 'C:/Users/RCHEN/Desktop/Rcode/title.txt', row.names = TRUE)

#grep("^A", year, value=TRUE)

#grep("^A", year, value=F)

##word <- c('abc noboby@stat.berkeley.edu','text with no email','first me@mything.com also you@yourspace.com')

##pattern <- '[-A-Za-z0-9\_.%]+@[-A-Za-z0-9\_.%]+\\.[A-Za-z]+'

##gregout <- gregexpr(pattern,word)