

PROJ 201 Project Final Report

Project Title:

Prerequisite Chain of Courses in Sabanci University

Name, Surname & ID of group members

Mete Harun Akçay, 27979

Emirhan Delican, 27820

Efe Üykü, 28409

Oğuz Çınar, 27984

Supervised by:

Hüsnü Yenigün

Month, day, year

05.19.2021

Abstract

This paper explains the main concepts of a project about visualizing the prerequisite chain of courses of Sabancı University, by reporting the methods and materials that have been used and discussing the results of the project. As mentioned in project proposal paper, the aim of the project was to code a website, in which students can enter the courses they want and see the whole prerequisite chain related to those courses. Accordingly, this paper discusses to what extent is the aim of proposal achieved and shows how it is achieved in detail. Additionally, the problems faced during the project and the solutions found to solve these problems are also discussed in this paper. Hopefully, it is expected that the product of this project, which is a well-designed website, will be used by Sabancı students in the future and it will make their course selection decisions easier.

Introduction

Almost all colleges have their own programs which require some number of courses for students to graduate. Additionally, these required courses usually have their own requirements, which creates a chain of program requirements for students. Sometimes this chain may be challenging and troublesome for students. Therefore, in this project, we gathered the prerequisite information of all courses and created the prerequisite chain of courses with the help of coding tools, which will be discussed in upcoming parts in more detail. Moreover, we implemented a program, which takes course names as input and displays all the prerequisite courses related to that course. Finally, we added this program to a website coded by us, which visualizes the output of the function in a way that it presents this complicated chain structure in an easier way, aiming that students can comprehend this chain effortlessly.

Methods & Materials

The very first job to be done in this project was creating the database which contains the information of prerequisites of courses. Python and two tools of Python were used in order to create the database. With the help of the tool called “BeautifulSoup”, all the links in Course Catalog, the page which contains the information of all courses in Sabancı University, were

scratched and added to a list in our Python program. However, some of the links were unnecessary for us; therefore, we made some changes in the list. For instance, links for recitation courses, link for returning the previous page, the link which leads us to go to Sabanci's main page were deleted from the list. After the selection is done, we utilized from another Python tool, which is "Requests", in order to display the content in the links. For the courses which have prerequisites, there was a part named as "Prerequisites". With the help of some basic coding skills, a new list was formed which contains the name of the courses and prerequisites of each course. If there was no prerequisite, "None" was written as prerequisite next to the course, and if there were logical expression like "and", "or" in the written prerequisite text, it was converted to a new form of representation ("&" for "and" and "|" for "or") and all the logical complications were avoided by using parentheses. After a txt file, in which each line contains the information of a different course, was created, we moved on to the next part of the project: writing an algorithm that reads all the information to visualize it in a website.

Since the second part was about requiring some web programming knowledge, the first thing to do in this part was getting familiar with the web programming languages such as html, css, bootstrap and php. Simultaneously, the graphical interface was being designed. After getting familiar with web programming concepts, the implementation was started with respect to the visual interface design. Homepage and two pages were coded accordingly. In both pages required information was explained briefly and was taken from the user as an html form input. Finally, the main algorithm that will hold the whole prerequisite information of the user input has been coded with the help of the txt file that was created in the first part of this project. Knowing the complexities in the logical expressions in this txt file was made algorithm design to be easier. The gathered prerequisite chain from the algorithm, needed to be turned into a graphical form in order to be understandable. For this purpose, "GraPHP" library was used. Firstly, the nodes which contains the related course names that was gathered by the algorithm were created, and then the necessary connections, which its info also gathered from the algorithm, between these nodes were made. In total, we observed that there are four different connections between the nodes, which will be discussed in results.

Results

As mentioned in the previous part, prerequisite information of each course was scrapped from Course Catalog and transformed in a way that it became more proper for programming language. Some examples of this transformation can be seen in Table1. As it can be seen by looking to table, unnecessary words are excluded, and new symbols are started to be used instead of logical expressions such as “and” and “or”.

Table 1.
The Transformation from Course Catalog to .txt File

Course	What is written below the word “Prerequisites” for the course, in Course Catalog:	What is written in the txt file that we have formed:
ANTH 214	(empty)	ANTH214: None
HIST 438	Undergraduate level ECON 202 Minimum Grade of D	HIST438: ECON202
MATH 317	Undergraduate level MATH 102 Minimum Grade of D and Undergraduate level MATH 201 Minimum Grade of D	MATH317: MATH102 & MATH201
ME 420	Undergraduate level ENS 207 Minimum Grade of D or (Undergraduate level ME 307 Minimum Grade of D and Undergraduate level ME 309 Minimum Grade of D) or (Undergraduate level ME 307 Minimum Grade of D and Undergraduate level ENS 202 Minimum Grade of D) or (Undergraduate level ME 309 Minimum Grade of D and Undergraduate level ENS 202 Minimum Grade of D)	ME420: ENS207 (ME307 & ME309) (ME307 & ENS202) (ME309 & ENS202)

After the process of database is done, the .txt files were sent to the group member who is responsible for coding the website, and the implementation has begun. The very first thing that was implemented was the homepage, which consists of info part and 2 different links which are directing some pages that will be implemented. (see Figure 1). Later that the pages which are directed by the links were implemented (see Figures 3, 4).

Figure 1. Homepage of the website

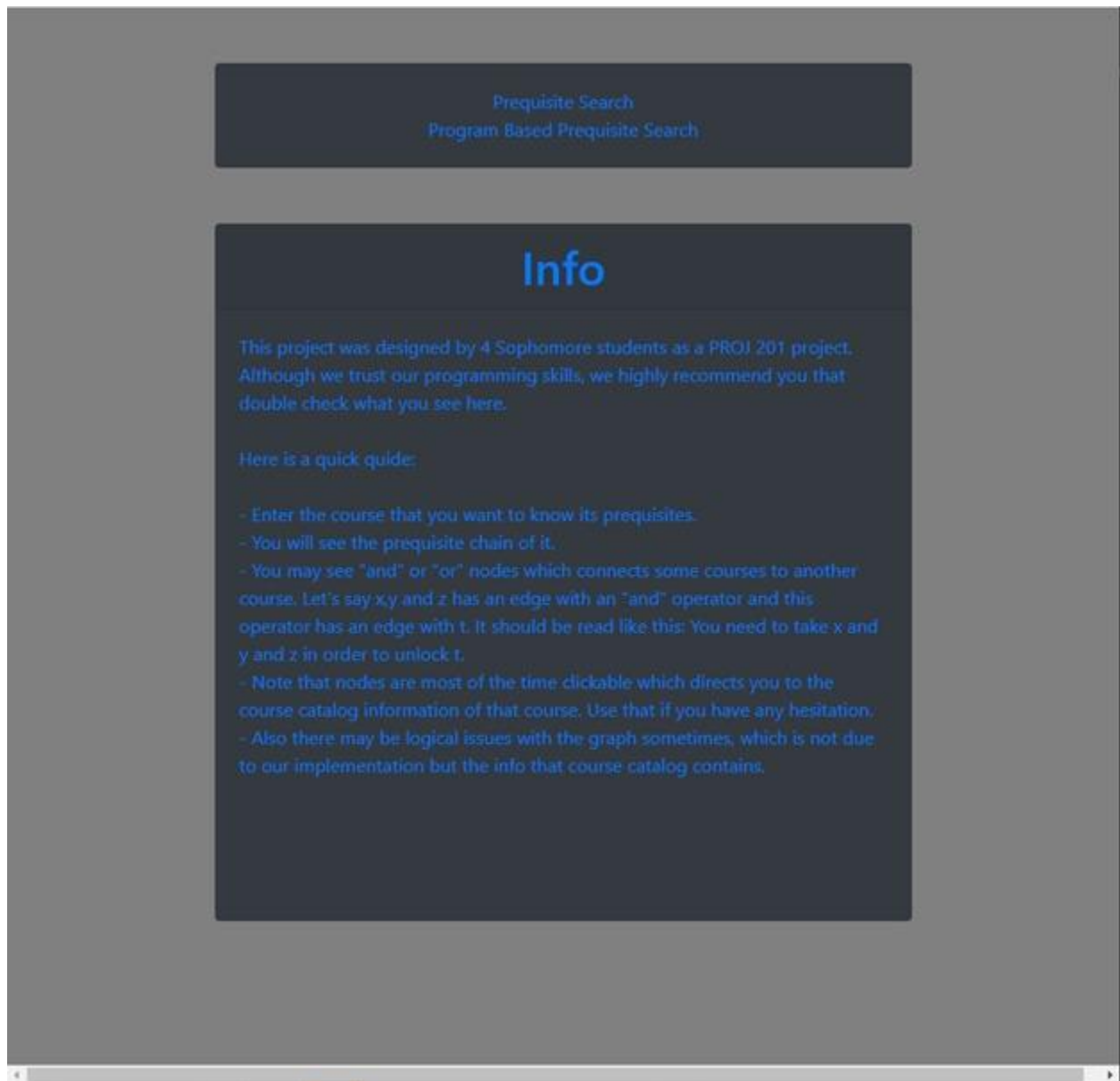
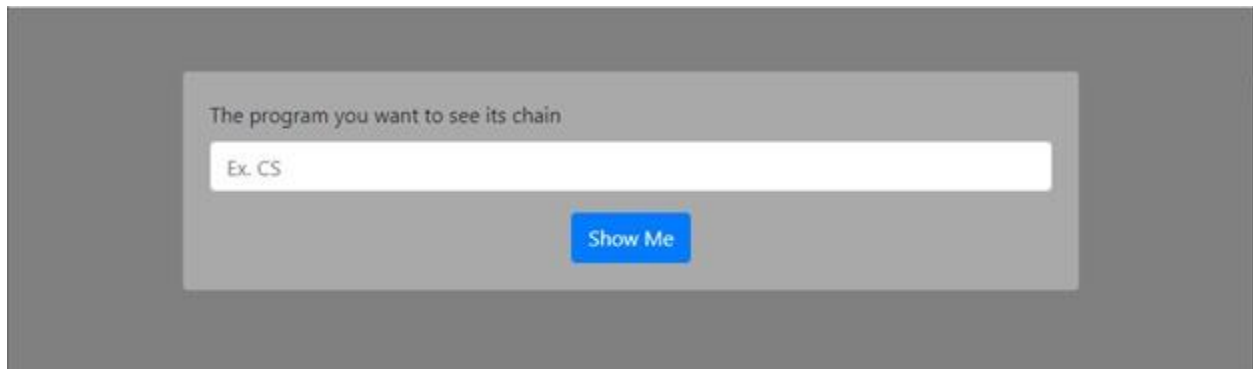


Figure 2. Interface of the Page of Program Based Prerequisite Search



The program you want to see its chain

Ex. CS

Show Me

Figure 3. Interface of the Page of Prerequisite Search



Course you want to know its prerequisites

Ex. CS204

Show Me

After the prerequisite chain implementation with the help of GraPHP library is done, we observe that there exist four different cases in prerequisite pathways of courses. The first and simplest one is the direct connection between the course nodes. For instance, if an X node points at Y node, it should be read as “in order to take Y, X must be taken” (see Figure 4). Another connection is two or more course nodes points at a newly created “and” node and this “and” node points at another course node. For example, if X, Y, Z nodes point at an “and” node, and this node points at Q node it should be read as “in order to take Q, X and Y and Z must be taken” (see Figure 5). In a similar way if X, Y, Z nodes point at an “or” node, and this node points at Q

node it should be read as “in order to take Q, one of the X or Y or Z must be taken (see Figure 6). Finally, the last connection type is where both “and”, “or” nodes are involved. This type of connections can be understood by utilizing all of the information above (see Figure 7). As a result, a program has been implemented in which all prerequisites related to the given input are clearly displayed, which can be seen in Figure 8.

Figure 4. Visualization of the case when only one course is needed to take the desired course.

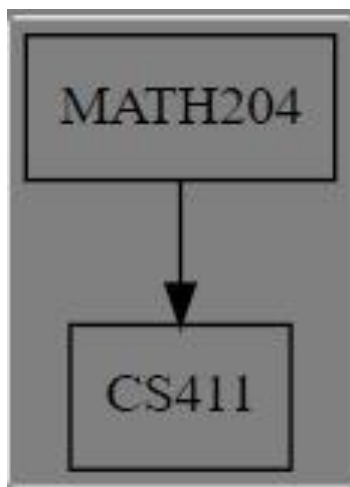


Figure 5. Visualization of the case when two or more courses are needed to take the desired course.

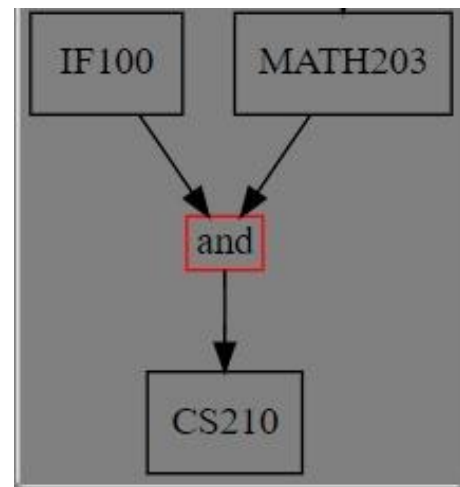
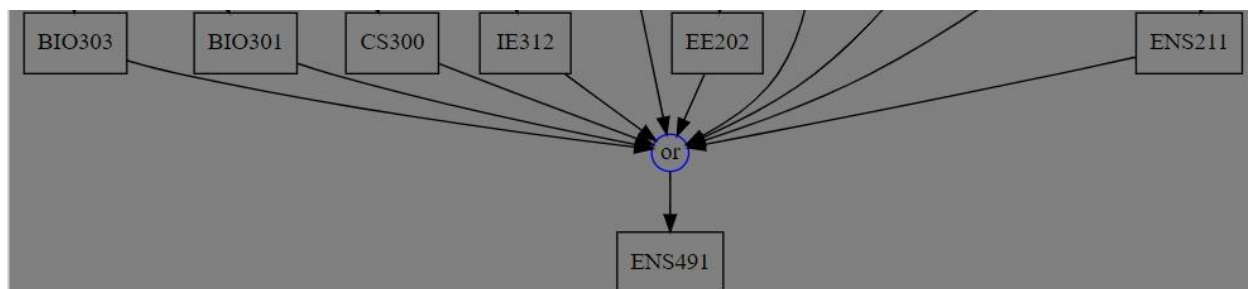


Figure 6. Visualization of the case when only one of the given courses is needed to take the desired course.



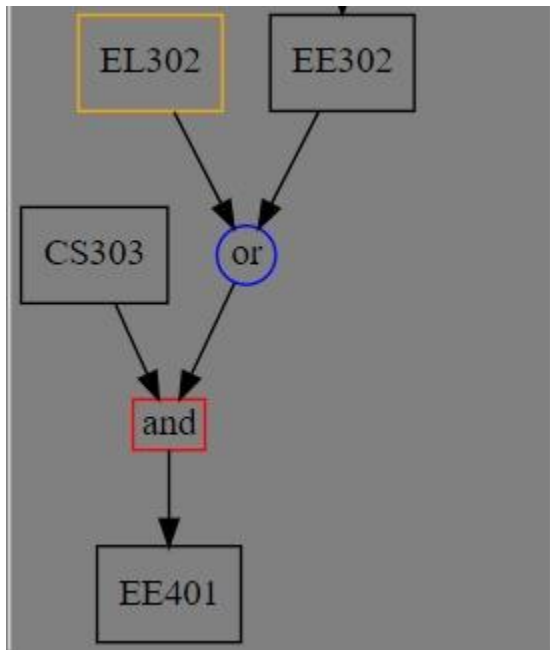
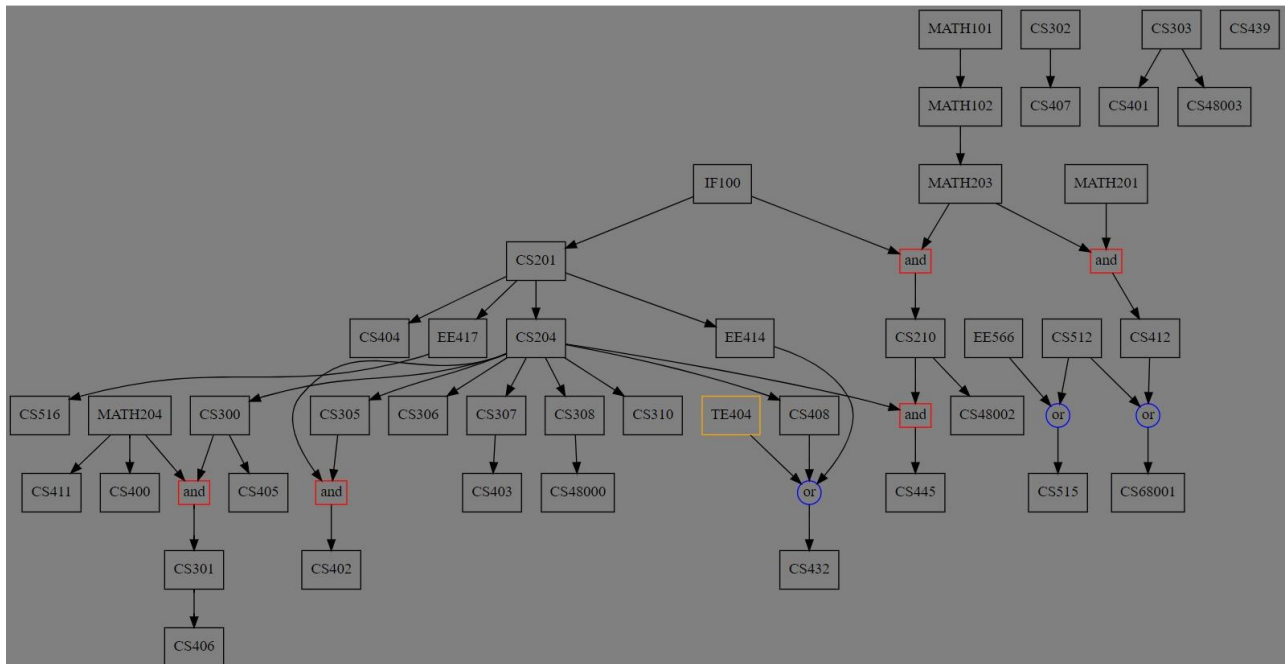


Figure 7. Visualization of the case when both “and” and “or” expressions exist.

Figure 8. Visualization of a complicated example. The output of the program when the input is “CS”



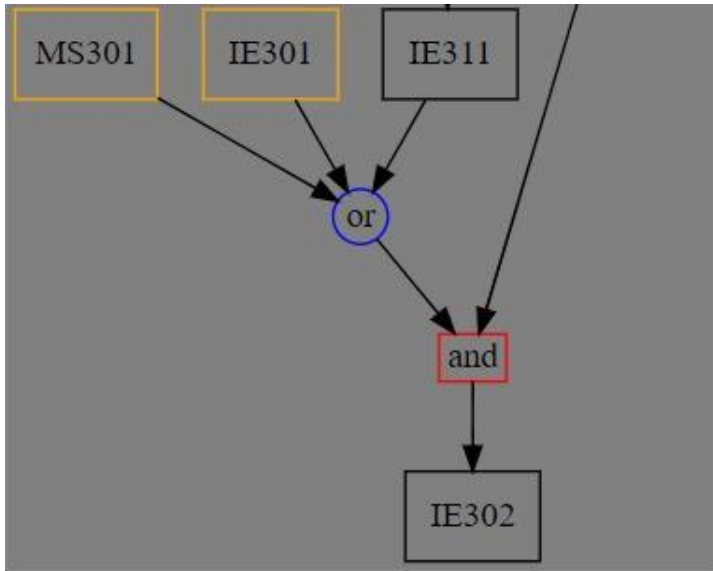
Discussion and Conclusion

During the project, some problems have been encountered and now they can be discussed. First of all, we realized that there are some courses, which are seen as prerequisite of some courses but not in the course catalog. To cope with this problem, one list for all prerequisite course and one list for all courses were created and compared. As a result, we found 76 courses that cannot be seen in course catalog although they are prerequisites of some of the courses. Additionally, we implemented our program in a way that these courses are in different color (see Appendix1) and students can notice them. These courses can be seen in Appendix2 and we suggest authorized people of Sabancı University to use this information to fix the problem.

Additionally, we realized that some of the courses have logical problems in their prerequisite information. For instance, if the prerequisite information of CS402 is examined, it can be seen that CS402 has 2 courses as its prerequisite, which are CS204 and CS305. However, when the prerequisite chain of CS305 is examined, it can be seen that CS204 has to be taken in order to take CS305. So, there is no need for CS204 in the prerequisite chain of CS402 because it is already a must to take CS305 (see Appendix 3). Most of the time the logical problems are very similar to this case, but there can be other types of logical issues which is not found by us. Although this problem was not in the scope of our project, we tried to find a way to detect and get rid of these unnecessary links. Unfortunately, we were not able to do that, and we highly suggest authorized people to work on this issue since it may create confusions among the students while they are taking courses.

As a conclusion, the program that we have implemented shows the whole prerequisite chain of a course which is entered as input. Therefore, it is not wrong to say that this project has achieved its purpose. The visualization of the program will help the students of Sabancı University in understanding and determining which courses they will select. A similar method of representation of data can be used also in other universities with the help of us. This program also revealed some errors in Course Catalog such as existence of courses which are shown as prerequisites of some courses, but not in catalog; or some logical mistakes/needlessness in chain; therefore, some necessary changes in Course Catalog can be done due to our project, in the future.

Appendices



Appendix1.

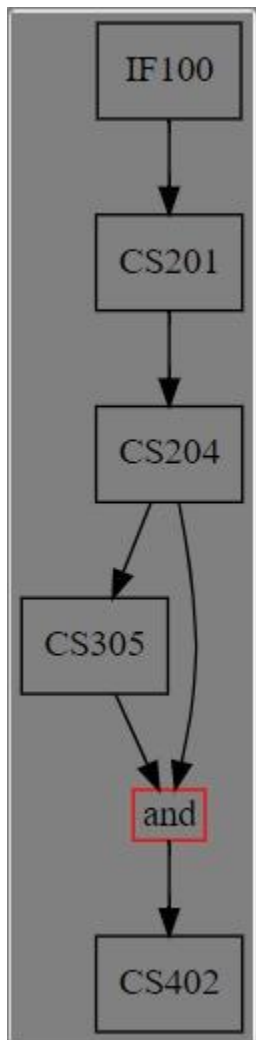
Visualization of “nonexistent” courses in the program.

(MS301 and IE301 do not exist in Course Catalog)

Appendix2.

List of the courses that are prerequisites of courses but not written in Course Catalog.

MGMT202	ARA301	ARA302	ARA501	ARA502	CH1102	TDP101	TDP102	TDP201	TDP202
TE404	CULT291	EL202	EL302	EL303	TE305	EL204	TE304	TE302	FRE102
FRE201	FRE202	CULT341	GER102	GER201	GER202	GER301	ENS555	HUM212	TLL501
TLL502	TLL520	HUM204	HUM214	MS301	IE301	MS303	MS309	MS304	MS302
MS305	MS401	ITA102	ITA201	JAP102	JAP201	JAP202	JAP540	KUR102	KUR201
KUR202	KUR502	KUR503	LAT312	LAT313	PERS301	PERS302	PERS303	PERS501	PERS502
PERS503	PSY303	PSY301	PSY370	PSY390	PSY352	RUS102	RUS201	SPA102	SPA201
SPA202	TLL301	TLL303	TLL501	TLL503	TE305				



Appendix 3:

An example of unnecessary prerequisite link

(The link between CS204 and CS402 is unnecessary since CS305 already requires CS204)

References

MySU Course Catalog. Retrieved May 15, 2021, from https://suis.sabanciuniv.edu/prod/bwckctlg.p_display_courses

Bibliography

BeautifulSoup Documentation. Retrieved March 19, 2021, from <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

Ronquillo, A. (2018). Python's Requests Library (Guide). Retrieved March 19, 2021, from <https://realpython.com/python-requests/>