**ENGR 102 Programming Practice**

**Mini Project 4**

*Posted on April 27, 2020, Due on May 11 2020 by 17:00*

*Last Update: April 27, 2020*

**Important Note 1: Please make sure that you are implementing your projects in Python 3 (submissions in Python 2 will not be graded!).**

**Important Note 2: Please do not use any extra libraries other than those specified in the project manual. Otherwise, your submission may be subject to additional grade cutoffs!**

You may take advantage of the functions that we covered in the class and available in [mysearchengine](https://lms.sehir.edu.tr/mod/resource/view.php?id=68579).py. Please do not copy and paste code from this file. Instead, import it into your files. [mysearchengine](https://lms.sehir.edu.tr/mod/resource/view.php?id=68579).py is available on LMS under week 11.

In this project, you are going to implement a search engine to search for professors in the College of Natural Sciences and Engineering (CENS) based on their academic profiles.

**What should it look like?**

The graphical user interface of your tool should look like the following one.

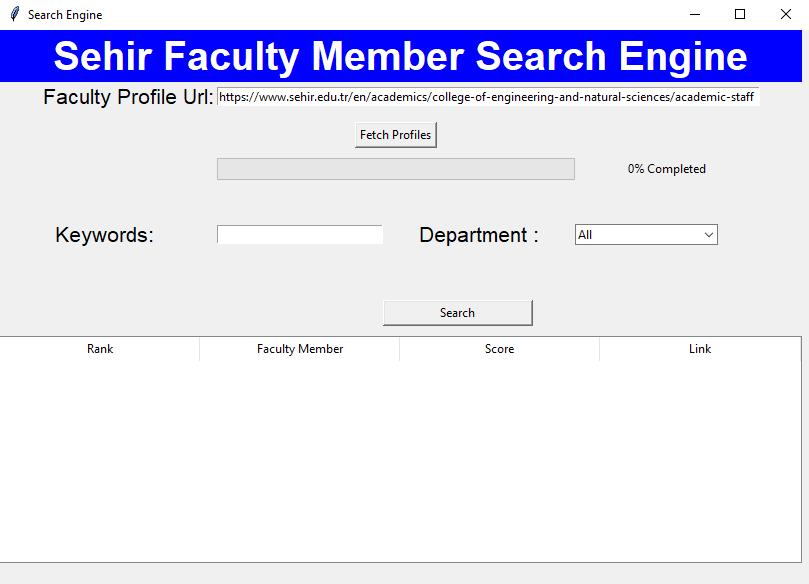


Figure 1: Initial view of GUI

**How should it work?**

Please see the animated gif posted on LMS to see a demo of how your application should work. Detailed instructions are provided below.

* There should be an entry widget at the top to get the link to the list of faculty members in CENS. By default, it should have the following URL when the program starts:

<https://www.sehir.edu.tr/en/academics/college-of-engineering-and-natural-sciences/academic-staff>

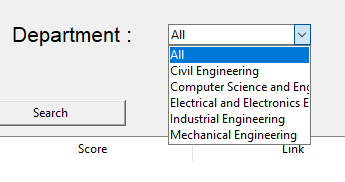
* When the “Fetch Profiles” button is clicked, the program should go to the url provided at the entry widget at the top[[1]](#footnote-0), and first collect the links to the individual profile pages of SEHIR Engineering faculty members, and then go to each link and extract the text in each faculty profile page. While your application is collecting links to faculty profile pages, you should notify the user with a label next to the progress bar, saying “Collecting Faculty Members’ Profile Links…” (see the following screenshot). At this step, you need to use selenium, since on Sehir website, the final form of HTML does not form until it is displayed by an internet browser so your application needs to open this website with a driver (i.e., chromedriver), and then request the HTML (please see the implementation notes and week 10 practice material for further details). A side effect of using selenium would be that the url that you are accessing will be automatically opened by a browser window in the background.



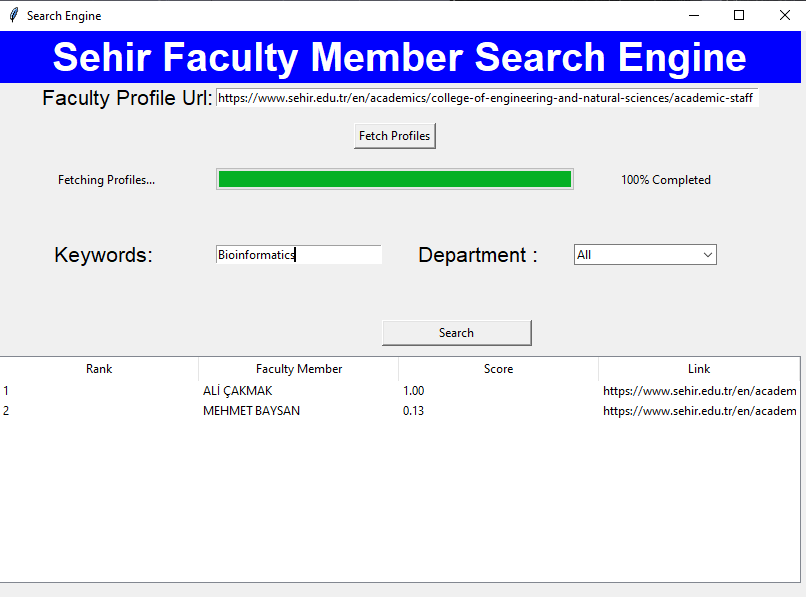
* Once the collection of personal profile page links is finished, your program will know how many profiles are available. Next, your program will go to each link and get their HTML context to parse the details about their academic interests and background. At this point, you don’t have to use selenium, you may simply use the requests module. When you have all details regarding each professor, you will need to implement a database (dbm or shelve) to record the frequency of each word that appears in faculty profiles.
  + While your application parses each professor you need to show your progress in the progress bar (see practice material of Week 4). That is, suppose that there are 50 profiles that you need to fetch. After completing the fetching of each profile, you will update the percentage of the completion accordingly (e.g., after fetching 10 profiles, you will show 20% completed in the progress bar and its label next to it). The label next to the progress bar at this stage should state “Fetching Profiles...”. (see the following screenshot).

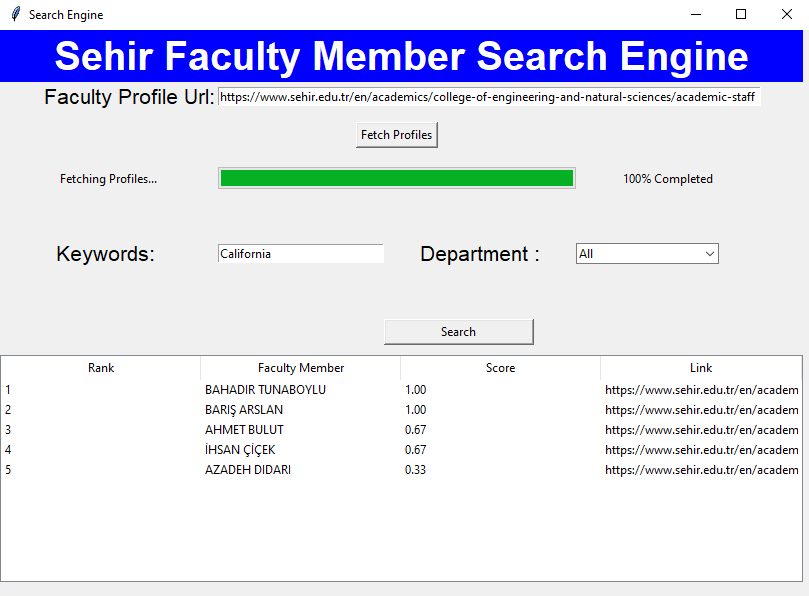


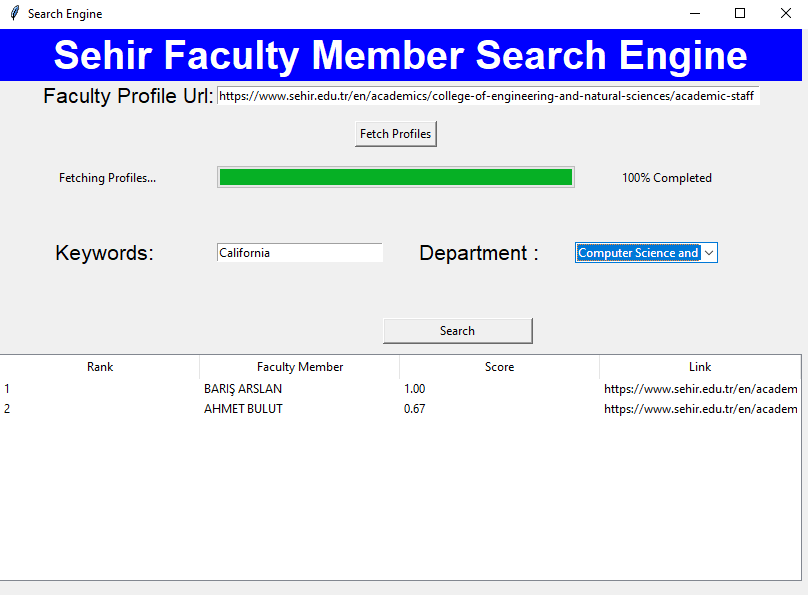
* + You must get department details of each professor and put them in the department combobox. Initially this combobox should be empty, and need to be filled later after parsing of all profiles finishes. In this combobox, there should not be any duplicates, and department names should be sorted in alphabetical order. (see the following screenshot). The default selection in this combobox should be “All” which will be added at the top.



* When the “Search” button is clicked, your program will get the keywords that are provided by the user and department limitations and will calculate a score for each professor that is in the provided department also has at least one of the keywords in their profile as recorded in the database. Their score will depend on the frequency of the keywords in their database, and the scores should be normalized as discussed in the class. (we will cover the scoring aspect in this or next week’s class).
* In the end when you have the scores and ranking for professors you will display them at the bottom in a Treeview widget. Treeview should have the following columns and corresponding values.
  + Rank: Rank (i.e., the order in the search results) of the professor calculated based on the given criteria.
  + Faculty Member: You should display their name
  + Score: Score that you calculated based on keywords and their frequencies
  + Link: Link to the professor’s profile page.
* After results are displayed correctly, when the user double clicks on the link on a professor’s link shown in the treeview’s link column, you should open the corresponding web page and show it to the user. (You may use selenium webdriver here as well. Please see the implementation notes).
* Here are some example screenshots from three different search sessions. Results were obtained by providing keywords “Bioinformatics” and “California” in three different sessions.







**Implementation Notes:**

* You may check Week 4 practice session materials to implement the progress bar.
* In order to be able to use selenium first you need to download a Chrome webdriver from the following website.
  + Chrome: <https://chromedriver.chromium.org/>
  + **Warning:** Your webdriver should be in the same folder with your code so that we’ll be able to run your code when we put a driver in your code folder since your code will be implemented in a way that it assumes the webdriver will be in the same folder.
* Once you downloaded the driver, the following piece of code will help you to open a page and get it’s outer HTML content, and then create a soup object on it if you like.

from selenium import webdriver

from bs4 import BeautifulSoup

driver = webdriver.Chrome(‘chromedriver.exe’)

driver.get('#Link to be opened by web driver)

elem = driver.find\_element\_by\_xpath("//\*")

html\_doc = elem.get\_attribute("outerHTML")

soup = BeautifulSoup(html\_doc, 'html.parser')

* For Treeview usage, please see the following examples:
  + <https://github.com/r2123b/tkinter-ttk-Treeview-Simple-Demo/blob/master/SimpleTreeview.py>
  + <https://riptutorial.com/tkinter/example/31880/treeview--basic-example>
* To be able to get the values of treeview rows clicked by the user you need to bind a method similar to the following code (This code is just an example and starting point don’t take it as it is).

def double\_click(self, event):

column = self.result\_treeview.identify\_column(event.x)

values\_row=self.treeview.item(self.treeview.focus())

# get the link url from treeview

# link = ...

driver = webdriver.Chrome(‘chromedriver.exe’)

driver.get(link)

* Column variable will give the number of the column that is clicked, if the user clicked the column with the link you’ll get the values of that row as shown with variable values\_row this variable will include the link that you want to open. Once you have the link you may pass it to the webdriver to open it.

**Warnings:**

* You **CANNOT** use the place for geometry, only the grid and packare allowed.
* Do not talk to your classmates on project topics when you are implementing your projects. Do not show or email your code to others. If you need help, talk to your TAs or myself, not to your classmates. If somebody asks you for help, explain them the lecture slides, but do not explain any project related topic or solution. Any similarity in your source codes will have serious consequences for both parties.
* Carefully read the project document, and pay special attention to sentences that involve “should”, “should not”, “do not”, and other underlined/bold font statements.
* If you use code from a resource (web site, book, etc.), make sure that you reference those resources at the top of your source code file in the form of comments. You should give details of which part of your code is from what resource. Failing to do so may result in plagiarism investigation. Last but not least, you need to understand code pieces that you may get some other resources. This is one of the goals of the mini projects.
* Even if you work as a group of two students, each member of the team should know every line of the code well. Hence, it is important to understand all the details in your submitted code.

**How and when do I submit my project?**

* Projects may be done individually or as a small group of two students (doing it individually is **strongly** recommended for the best learning experience). If you are doing it as a group, only **one** of the members should submit the project. File name will tell us group members (Please see the next item for file naming details).
* Submit your own code in a single Python file. Name it with your and your partner’s first and last names. As an example, if your team members are Deniz Barış and Ahmet Çalışkan, then name your code file as deniz\_baris\_ahmet\_caliskan.py (Do not use any Turkish characters in the file name). If you are doing the project alone, then name it with your name and last name similar to the above naming scheme.
  + Those who do not follow the above naming conventions will **get** 10% **off** of their project grade.
* Submit it online on LMS by 17:00 on the due date.

**Late Submission Policy:**

* -10%: Submissions between 17:01 – 18:00 on the due date
* -20%: Submissions between 18:01 – midnight (00:00) on the due date
* -30%: Submissions after which are up to 24 hours late.
* -50%: Submissions which are up to 48 hours late.
* Submissions more than 48 hours late will not be accepted.

**Grading Criteria?**

|  |  |  |  |
| --- | --- | --- | --- |
| **GUI**  **25 pts** | **Fetching Profiles Properly and Saving them to a Database**  **30 pts** | **Progress Bar Works Properly**  **20 pts** | **Search Works Properly with All Filters, Ranking, Score Computation, etc.**  **25 pts** |

From your overall grade, we will deduct points by the specified percentage for the following items:

* + Inappropriate/cryptic variable names (-10%)
  + Classes and objects are not used properly (-20%)
  + Insufficient commenting (-10%).
  + Inappropriate file naming (-10%)

**Have further questions?:**

If you need help with anything, please send an email to class TAs. You may make an appointment to have a 1-to-1 Google Meet meeting if needed.

1. <https://www.sehir.edu.tr/en/academics/college-of-engineering-and-natural-sciences/academic-staff> [↑](#footnote-ref-0)