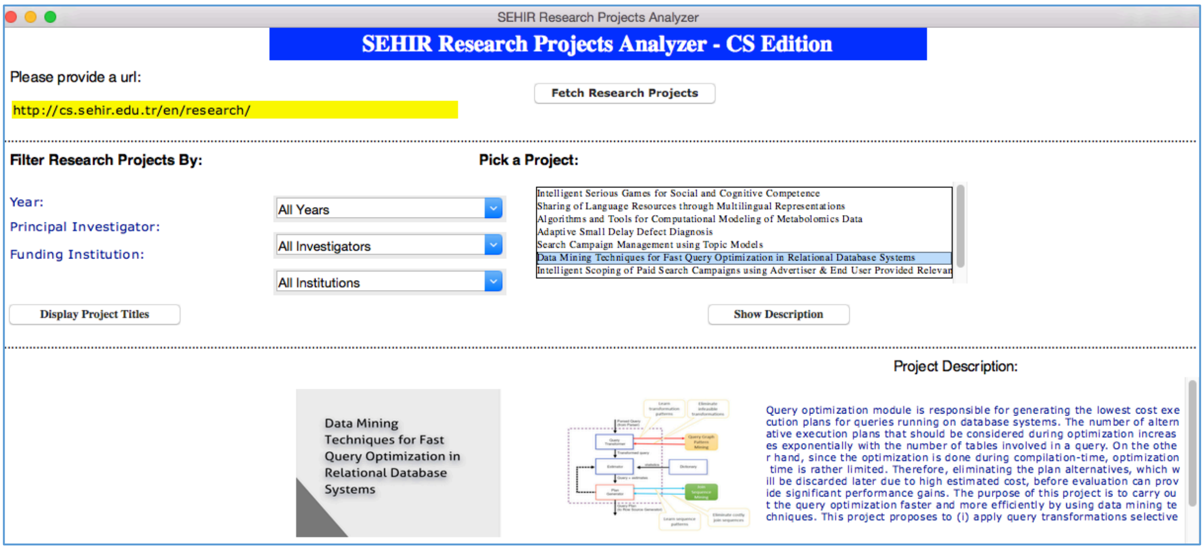
ENGR 212 Programming Practice
Mini Project 4

April 12, 2016

*(**Important:** Please note the deadline day change from Tuesday to Monday)*

The research projects conducted by faculty members at a university are usually published on university web pages. As an example, research projects done at SEHIR's Department of Computer Science & Engineering are published at http://cs.sehir.edu.tr/en/research/. In this project, you are going to develop a tool that will fetch information on research projects conducted by CS faculty members, and allow performing simple analyses on the fetched information. Details regarding the requirements are as follows:

1. Your program will have a graphical user interface (GUI) which will look like as shown in Figure 1. Details about how it should work are provided below.

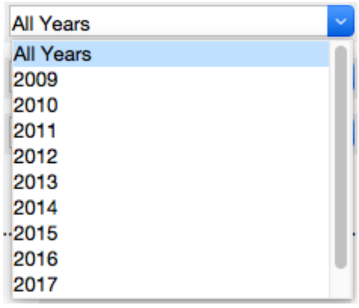

SEHIR Research Projects Analyzer
SEHIR Research Projects Analyzer - CS Edition
Please provide a url:
http://cs.sehir.edu.tr/en/research/
Fetch Research Projects
Filter Research Projects By:
Pick a Project:
Year:
Principal Investigator:
Funding Institution:
All Years
All Investigators
All Institutions
Display Project Titles
Show Description
Project Description:
Data Mining Techniques for Fast Query Optimization in Relational Database Systems

Intelligent Serious Games for Social and Cognitive Competence
Sharing of Language Resources through Multilingual Representations
Algorithms and Tools for Computational Modeling of Metabolomics Data
Adaptive Small Delay Defect Diagnosis
Search Campaign Management using Topic Models
Data Mining Techniques for Fast Query Optimization in Relational Database Systems
Intelligent Scoping of Paid Search Campaigns using Advertiser & End User Provided Relevan

Query optimization module is responsible for generating the lowest cost execution plans for queries running on database systems. The number of alternative execution plans that should be considered during optimization increases exponentially with the number of tables involved in a query. On the other hand, since the optimization is done during compilation-time, optimization time is rather limited. Therefore, eliminating the plan alternatives, which will be discarded later due to high estimated cost, before evaluation can provide significant performance gains. The purpose of this project is to carry out the query optimization faster and more efficiently by using data mining techniques. This project proposes to (i) apply query transformations selective

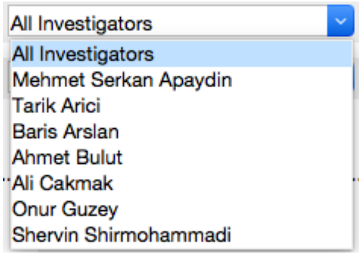
Figure 1

- The user will provide a URL to a web page that contains information on CS research projects. Then the user will click on "Fetch Research Projects" button.
- Then, your tool will process the web page at the provided url using urllib2 and BeautifulSoup, and fill in the "Year", "Principal Investigator", and "Funding Institution" combo boxes with proper values. More specifically, "Year" combo box will list all years that are between the minimum project start year and maximum project end year found at the processed web page "Principal Investigator" combo box will list all CS faculty members (sorted by last name) who has at least one project listed on the web page. "Funding Institution" combo box should list all possible funding institutions that are listed on the project web page. The default selected value in all combo boxes should initially be "All". (see the screenshot below).



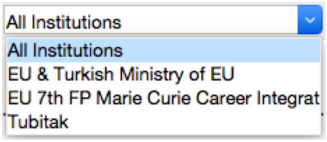
All Years
All Years
2009
2010
2011
2012
2013
2014
2015
2016
2017

Year combo box entries



All Investigators
All Investigators
Mehmet Serkan Apaydin
Tarik Arici
Baris Arslan
Ahmet Bulut
Ali Cakmak
Onur Guzey
Shervin Shirmohammadi

Principal Investigator combo box entries



All Institutions
All Institutions
EU & Turkish Ministry of EU
EU 7th FP Marie Curie Career Integrat
Tubitak

Funding Institution combo box entries

- Next, the user may select particular entries in some combo boxes (and/or leave some or all of them as "All"). Then, the user will click on "Display Project Titles" button. According to user's selections in the combo boxes, your application will list a subset of all available projects in a list box on the right. The list box should have a vertical scroll bar to accommodate large project lists. For instance, let's say that the user has selected 2013 in the first combo box, 'Ahmet Bulut' in the second combo box, and "Tubitak" in the third combo box. Then, you should list projects for which start_year <= 2013 <=end year, principal investigator is "Ahmet Bulut", and funded by "Tubitak". If the user leaves a combo box with its default value (i.e., "All"), then that field will not have any filtering effect on the final project list.

- Then, among the listed projects in the list box, the user will select a project title, and for the selected project title, your program will display its poster picture and summary at the bottom part as shown in Figure 1. The summary area should have a vertical scroll bar so that if the project summary does not fit to the provided area, then the user can scroll down to see the remaining parts.

Can you provide any further pointers that may be helpful? :

- To display the poster picture of a project, you may initially just extract the picture url from the page source by using BeautifulSoup first at the initial parsing phase, and record it in a dictionary-like structure. If a user selects a particular project from the listbox, then you can get the url of the poster picture for that project from where the poster url dictionary, and download the picture at that time. To download and display a picture on GUI, you may check the following example:

  https://www.daniweb.com/programming/software-development/code/440946/display-an-image-from-a-url-tkinter-python

Warnings:

- **Do not** talk to your classmates on project topics when you are implementing your projects. **Do not** show or email your code to others. **Do not** work together if you are not in the same group. 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 resource 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.

- 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. You may be interviewed about any part of your 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 recommended for 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 details).

- Submit your own code in a **single** Python file (Do **not** include other files that you import). Name it with your and your partner's first and last names (see below for naming).
  - 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 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 -5 off** of their project grade.
- Submit it online on LMS (Go to the Assignments Tab) by **17:00 on Monday, April 25, 2016 (Please note the Monday submission, it is not on Tuesday)**.

  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 which are 24 hour late.
  - -50%: Submissions which are 48 hours late.
  - Submission more than 48 hours late will not be accepted.

Grading Criteria? :

Code Organization			Functionality					
Meaningful variable names (%3)	Classes and objects used (%4)	Sufficient commenting (%4)	Compiles? (20)	GUI Design (10)	Reading project info / Populating combo boxes (20)	Filtering out projects based on different user selections (20)	Fetching and displaying project poster properly (10)	Displaying the selected Project summary properly (10)

- Interview evaluation (your grade from interview will be between 0 and 1, and it will be used as a coefficient to compute your final grade. For instance, if your initial grade was 80 before the interview, and your interview grade is 0.5, then your final grade will be 80*0.5 = 40). Not showing up for the interview appointment will **result in** grade 0.

Have further questions? :

- Please contact your TAs (Jareth or Dogukan are focusing on projects. You may want to talk to them first, but you may talk to Ali and Bekir as well) if you have further questions. If you need help with anything, please use the office hours of your TAs and the instructor to get help. **Do not walk in randomly (especially on the last day) into your TAs' or the instructor's offices. Make an appointment first. This is important. Your TAs have other responsibilities. Please respect their personal schedules!**