1. What are the names and NetIDs of all your team members? Who is the captain? The captain will have more administrative duties than team members.
    a. Team Members:
        i. Bashir Partovi (partovi2@) - Captain
        ii. Karthik Rajagopal (kr22@)
        iii. Mohana Venkata Kalyan Cheerla (cheerla3@)
2. What system have you chosen? Which subtopic(s) under the system?
   Topic: Improving Expert Search
   Subtopics: Automatic web crawler, topic modeling of bio pages, and faculty page classification

   We have chosen to improve upon Expert Search. We plan to add a faculty page classifier module, topic modeling for bio pages, and an automatic web crawler. Faculty page classifier integrates with the automatic web crawler to detect faculty pages, given a university URL. The crawler will run on scheduled bases and dynamically updates the indexer for the front-end UI. Hence, since we do not store bios statically, we are able to obtain an updated list of faculties and their bios; a feature that the old system lacks. In addition, by doing topic modeling on the bios, we can tag each search result with top relevant skills.

3. Briefly describe the datasets, algorithms or techniques you plan to use
    a. Faculty page classification - we have a huge resource of faculty directory page URLs available in the sign-up sheet to serve as our positive examples. In addition, we will be using a list of random URLs as our negative examples to train a model to classify URLs
    b. Topic modeling for bio pages - We will be using spaCy library to do topic modeling on bio pages and include it in the search results as separate tags
    c. Automatic Web Crawler - we will be developing an automatic web crawler that uses BFS to crawl the university pages in search of faculty URLs. This module will be integrated with the classifier to index faculty pages
4. If you are adding a function, how will you demonstrate that it works as expected? If you are improving a function, how will you show your implementation actually works better?
    a. Faculty Page Classifier - we will split the existing compiled list of faculty pages into two separate datasets, one for training the classifier and one for testing/validating it. Based on that, we will be able to create a confusion matrix that shows the effectiveness of our classifier
    b. Automatic Web Crawler - We will run the crawler on the university pages mentioned in the sign-up sheet to see what percentage of bio pages are found during crawling. Based on that, we will be able to assess whether the crawler is able to reach all faculty pages from the main URL.
    c. Topic modeling of bio pages - We will run the topic modeler on a small selection of bio pages we prepared and based on the extracted topics, we will be able to verify manually whether the topic modeler works correctly

5. How will your code communicate with or utilize the system? It is also fine to build your own systems, just please state your plan clearly

   We will be improving the existing system. The webcrawler runs as a separate process and continuously crawls the web pages of the universities mentioned in the sign-up sheet. Using the faculty page classifier, it is able to store bio pages and run information extraction scripts. In addition, we will be changing the UI in order to add the modeled topics for the given search results.
6. Which programming language do you plan to use?
   Python
7. Please justify that the workload of your topic is at least 20*N hours, N being the total number of students in your team. You may list the main tasks to be completed, and the estimated time cost for each task.

Task	Time
Topic Modeler - Research/Implementation Since this is a new subject for the team, research and understanding of the subject is time consuming	9 Hours
Topic Modeler - Test Data Preparation We also have to manually prepare test cases for our topic modeler.	5 Hours
Topic Modeler - Tuning This is allocated to fix any bug or issue we experience during testing and making sure the modeled topics are accurate	4 Hours
Topic Modeler - UI Integration We need to change the UI to include the modeled topic for the bios that show up in the search results	2 Hours
Faculty Page Classifier - Research/Implementation None of us worked on any project that involves training a model to do classification; therefore, this is a huge learning curve for us	16 Hours
Faculty Page Classifier - Training Data/Test Data Preparation We need to create both the training and the test data for our model	5 Hours
Faculty Page Classifier - Tuning the	3 Hours

model This is to address any issue that we may encounter during the classification	
Web Crawler - Implementation	8 Hours
Web Crawler - Testing We need to make sure that the crawler is actually visiting the faculty pages by looking at the visited URLs	5 Hours
Web Crawler - Performance Tuning It could take a long time for a crawler to finish crawling. We need to be able to find performance bottlenecks and remove them	4 Hours
Overall Documentation	8 Hours