**Project Overview**

For the project, you will work in teams of **up to two students** on a problem of your choosing that is interesting, significant, and relevant to Information Storage & Retrieval. You will have great latitude in what you choose to work on, so take advantage of this opportunity to make a big impact!

To give you some ideas on potential topics, in the previous classes, people have worked on various topics, including *opinion based search for social media platforms, rank documents for job appliers based on skill set, CSE Department Faculty Search Engine, Video Game Searcher, Food Recipe Retrieval, movie search, music search, personalized search* etc.

The primary requirements of the project are:

- Your project code must live on github. We prefer it to be public. However, if you're too scared to share your code with future employers, you can claim a private github account (with a .edu email address).
- Your project must use some non-trivial data that your team collects. You may choose to sample social media data (e.g., from one of the APIs listed over at Programmable Web), download an existing collection (e.g., Mind, Wikipedia, IMDB), write a simple crawler to collect pre-organized data (e.g., CIA docs, The Simpsons, or write your own custom web crawler.
- Your project must make a sufficient technical and/or intellectual contribution. You may re-use existing toolkits, libraries, and so forth. But you should demonstrate some depth to your project.
- Your project must be deployed as a web app or mobile app.

**Grading Criteria**

The course project counts for 25% of your final grade. You will receive an overall rating based on the performance of your entire team, as well as an individual rating based on the feedback of your teammate(s). Typically, the individual rating can bump or depress your project grade by some small delta (say moving a group rating of 85/100 plus or minus 5 points). A project score may be depressed significantly if a group member makes only a superficial contribution to a project. Your project will be graded based on the following key milestones:

- [10%] Project proposal, Due Oct. 7 by 11:59pm;
- [10%] Checkpoint 1: Data, Due Oct. 21 by 11:59pm;
- [10%] Checkpoint 2: Core Algorithm, Due Nov. 4 by 11:59pm;
- [10%] Checkpoint 3: UI, Due Nov. 11 by 11:59pm;
- [10%] System Deployment and Presentation Slides, Due Nov. 20 by **noon (Note: not end of day)**;

- [30%] Project presentation in class, on Nov. 21 (Th) and Nov. 26 (Tu);
- [20%] Written Report, Due May 1st by 11:59pm.

Recall that your late days are applicable to homework assignments only. All project milestones are due on their respective due date. No late project milestones will be accepted.

==========

**Project proposal** [1 to 2 pages (PDF); Post on Canvas]

Each group should post a 1-2 page project proposal in PDF to the Canvas course discussion forum. Be sure to start a new thread for your proposal and name the thread "Proposal: [project_name]", where [project_name] is a brief, descriptive name of your project. Your name should be something memorable!

In the proposal, you should address the following issues (adopted from C. Zhai):

- What is exactly the function of your tool? That is, what will it do?
- Why would we need such a tool and who would you expect to use it and benefit from it?
- Does this kind of tools already exist? If similar tools exist, how is your tool different from them? Would people care about the difference? How hard is it to build such a tool? What is the challenge?
- How do you plan to build it? You should mention the data you will use and the core algorithm that you will implement.
- What existing resources can you use?
- How will you demonstrate the usefulness of your tool?

==========

**Checkpoint 1: Data** [1 page MAX (PDF); Post on Canvas]

For the first project checkpoint, you must have collected a significant portion of the data that your project will ultimately use. You should describe the data in 1 page MAX that you will be using for your project. You should show figures highlighting the key characteristics of the data. For example, you might show us the distribution of the lengths of web pages you have encountered. Your goal here is to convince us that you have actually spent the time to collect your data and that you have a solid grasp on manipulating it. Post your 1-page PDF to the same thread as your original proposal.

==========

**Checkpoint 2: Core Algorithm** [Release on github]

For the second project checkpoint, you will release on github a working partial implementation of your project. By this checkpoint, you must have implemented at least your core algorithm. Your release should include simple test code that shows your algorithm is working over some small set of data. Our expectation is to see -- at a minimum -- code along the same lines as in

your past programming assignments. You should prepare a brief README file to tell us how to set up our environment to run your code (e.g., are there dependencies we should be aware of?).

==========

**Checkpoint 3: UI** [2 pages MAX (PDF); Post on Canvas]

For the third project checkpoint, you will post a 2-page PDF of your user interface and interaction design. We expect to see actual screenshots. You may supplement your screenshots with sketches of how the user interaction will be supported, but we will want to see that you have actually prepared a prototype UI.

==========

**System Deployment and Slides** [Post a link on Canvas, Post slides on Canvas]

You need to deploy your system as a web app or a mobile app. Post a link to your app on Canvas. Prepare your slides for in-class project presentation and post slides on Canvas.

==========

**Project presentation in class**

You will give a short presentation of around 5-10 minutes in class. We will decide the exact length for project presentations later depending on how many projects we have.

==========

**Written Report** [10 pages MAX (PDF); Post on Canvas]

Post a written project report on Canvas. In this report, you should first give a brief introduction to the system you built, describe the problem it is addressing and its main functions, then describe the overall approach, provide enough implementation details, describe the dataset you crawled or used, describe evaluation settings, results, result/error analysis and conclusions.
==========