CS 410: Text Information Systems  
Group Project Documentation  
Team Green Koalas

Team Members:

* Yuxiang Huang - [yh43@illinois.edu](mailto:yh43@illinois.edu)
* Bob Manasco - [manasco2@illinois.edu](mailto:manasco2@illinois.edu) - Group Coordinator
* Cullen Stone - [cpstone2@illinois.edu](mailto:cpstone2@illinois.edu)

Project Installation:

*The following instructions have been tested with Python2.7 on Linux and MacOS*

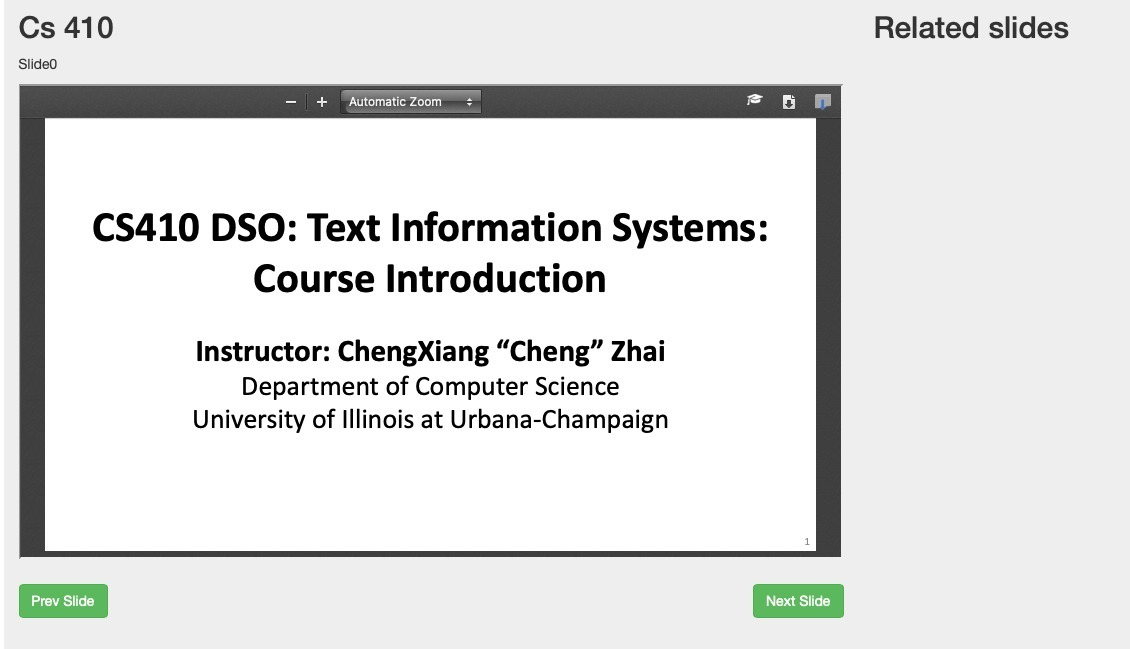
1. Source code (including Team Green Koala enhancements) can be found at https://github.com/CS410Fall2020/CourseProject
2. You should have ElasticSearch installed and running --  
   https://www.elastic.co/guide/en/elasticsearch/reference/current/targz.html
3. Create the index in ElasticSearch by running python create\_es\_index.py from EducationalWeb/
4. Download tfidf\_outputs.zip from here -- https://drive.google.com/file/d/19ia7CqaHnW3KKxASbnfs2clqRIgdTFiw/view?usp=sharing
5. Unzip the file and place the folder under EducationalWeb/static
6. Download cs410.zip from here -- https://drive.google.com/file/d/1Xiw9oSavOOeJsy\_SIiIxPf4aqsuyuuh6/view?usp=sharing
7. Unzip the file and place the folder under EducationalWeb/pdf.js/static/slides/
8. From EducationalWeb/pdf.js/build/generic/web, run the following command: gulp server
9. In another terminal window, run python app.py from EducationalWeb/
10. The site should then be available at http://localhost:8096/

How to use:

The EducationalWeb system was developed by some of the students in Prof. Zhai's research group for navigating through course slides. At present, it only contains the slides for CS410, but could be expanded in the future to include other courses.

Below are some of the features of the EducationalWeb tool:

* Choosing a lecture using the drop-down list in the navigation bar
* Sequentially navigating through the lectures/slides using the Next and Prev buttons at the bottom of a slide
* Searching for relevant slides using the search bar at the bottom of a slide
* Navigating to a related/recommend slide from the column on the right
* Finding an explanation of a term/phrase on the slide by highlighting it and then clicking on the "Explain selected text" button on the top-right of a slide. It will try to retrieve a relevant section from Professor's textbook that contains an explanation of the selected phrase.
* Downloading the current slide being viewed by clicking on the “Download” button, to the right of the “Explain selected text” button.
* Downloading all slides for the lecture for the current slide being viewed by clicking on the “Download All” button, to the right of the “Download” button.  
  (\*\*\*NOTE: this is the function that was added by Team Green Koalas\*\*\*)



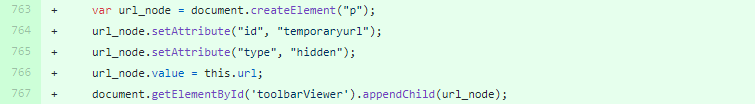
Download

Explain selected text

Download All

Implementation:

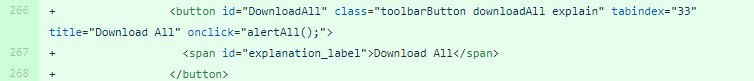
First, in the /build/generic/web/viewer.js file, we included some code that allows us to see the url for the slide currently being viewed.



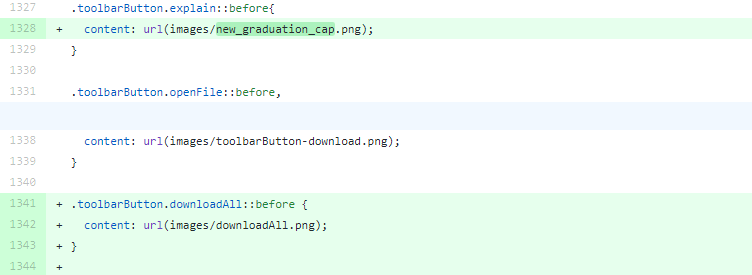
Then, in the /build/generic/web/viewer.html file, we updated some code and wrote several new functions:

* alertAll() is used to get the current-slide URL we provided in viewer.js (above) and convert it to a usable format for our purposes.
* getURLs() converts the formatted current-slide URL into a list of potential slides that may be contained within that file’s folder.
* zipItUp() takes that list of potential file names and adds any existing files to a zip file (“slide-series.ip”) and then provides the opportunity for a user to save that zip file through their browser.
* We also added a new button “Download All” to initiate this function.





Finally, in the /build/generic/web/viewer.css file, we made changes necessary to format the “Download All” button referenced above. We also fixed the graphic for the existing button for the “Explain Selected Text” button to allow it to fit properly.



Justification:

Our intention was to also take on one additional improvement (performance, enabling new courses, etc.), but ran into several impediments that affected our ability to do so. First, setup of the EducationalWeb environment took much, MUCH longer than anticipated. One team member was tracking his time and can show (in Timesheet.xlxs) that he spent over 10 hours just trying to configure the system. Most of the problems were caused either by attempting to set up the tool in a Windows environment or using a newer version of Python. Another team member finally had success on a Mac, so the third team member borrowed a Mac computer, which allowed him to finish setup relatively quickly after that.

Also, we did not realize when taking on this project that the work that needed to be done was almost entirely in Javascript. One team member has no experience in Javascript, and the other two have only slight knowledge, so it took a while just to get up-to-speed with this technology. If we had known more, we would not have been so cavalier in just assuming we’d be able to zip an entire folder at once or else get a list of files from a server folder on a client Javascript – which we now know we cannot do. MUCH time was spent researching ways to do this, including Node.js, JQuery, PHP, Ajax, and Kintone. We finally reverted to just using the file naming convention in our favor and assuming there would be fewer than 99 files in any lecture directory.

According to the Timesheet.xlsx file, one team member spent over 30 hours on this project, which exceeds the requirements for this assignment, so we did not address any issue beyond the multiple-slide download enhancement.