Sean Lai

Course Project

CS410

Documentation

**Summary:**

This project is to improve the functionality of to save next slides in memory so users do not need to load each new slide every click. I have implemented this feature by using in-memory list within python when server is running, to load multiple slide pages at once, and as a queue, the system iterate through the list when the user clicks “Next” button on the EducationalWeb page.

In the original, the EducationalWeb system would resolve each slide at the moment of user click, which would only allow a page to load one at a time, causing delays and negative user experience.

**Implementation:**

I have created a new function on app.py, which is the main function python file for the system, called `buffer\_new\_slides`. This function will iterate a number of slides, with number determined by the global variable SLIDE\_BUFFER\_SIZE, to fill up a python list SLIDE\_BUFFER\_LIST with generated slide renders. With this, users would experience a longer load when they first click through the slides, as the system would buffer multiple slide renders at once and save it to memory, but further iteration through the slides will be very fast as it’s in memory.

The list will work like a queue, so whenever a slide is read and ready to move on, the rendered slide is removed from the list, and a new one will be buffered for the user. This allows an in-memory speed of slide viewing experience for the user, instead of must loading a rendered slide object with `resolve\_slides`, on every click.

Please see my video demonstration for the code and operation for more details.

**Challenges:**

Because I did not a have a web server to test this on, the localhost server proved difficult to test if my change really made the slide viewing faster. This is due to the fact that original slide viewing was already fast enough since all the slide files were local. However, it is very small difference with speed with new change, and I expect it to be much more noticeable when the slides are up on a web server.

**Out of Scope:**

This implementation would only work on “Next” slide button, not going backward in pages with “Prev”. The Prev button would work, however would mess up the order of the list. This can be fixed by implementing similar queuing system for Prev in the same way as Next, however due to time constraints I am only going to demo for Next button.

**How to use the software:**

*# Documentation/Instructions*The following instructions have been tested with Python2.7 on Linux and MacOS  
  
1. You should have ElasticSearch installed and running -- https://www.elastic.co/guide/en/elasticsearch/reference/current/targz.html  
  
2. Create the index in ElasticSearch by running `python create\_es\_index.py` from `EducationalWeb/`  
  
3. Download tfidf\_outputs.zip from here -- https://drive.google.com/file/d/19ia7CqaHnW3KKxASbnfs2clqRIgdTFiw/view?usp=sharing  
   
 Unzip the file and place the folder under `EducationalWeb/static`  
  
4. Download cs410.zip from here -- https://drive.google.com/file/d/1Xiw9oSavOOeJsy\_SIiIxPf4aqsuyuuh6/view?usp=sharing  
   
 Unzip the file and place the folder under `EducationalWeb/pdf.js/static/slides/`  
   
5. From `EducationalWeb/pdf.js/build/generic/web` , run the following command: `gulp server`  
  
6. On line 38 on `EducationalWeb/app.py`, edit the value here to an Integer for "number of slides to buffer at a time".  
   
 I have set the default to "5", so you can also leave it as is.  
  
7. In another terminal window, run `python app.py` from `EducationalWeb/`  
  
8. The site should be available at http://localhost:8096/