ISTE-610 Knowledge Representation Technologies
Twitter Project
Fall 2015

Due Dates

12/7/2015 11:59PM. Files must be submitted to dropbox in myCourses; feel free to submit work in progress. Try to maintain the same filename for different versions of the same work. myCourses will notify the instructor as to which file is most recent. Each team member must submit all the files. Any discrepancy between team member submissions will raise questions about the participation of each team member. Bear in mind that failure to submit files to the dropbox can result in a failing grade and that ridiculous excuses like "we're just trying to make it easier for you" are worn out. All submissions must bear the names of all team members.

12/8 and 12/10/2015. Demos in class as time permits, any remaining demos to be completed in instructor's office.

Description

In this assignment, you will develop an application using MongoDB. The goal of the assignment is to develop a basic search application that can be used to search tweets or comparable data of your choice.

Backend Application

For backend processing you must use the MongoDB database management system. You should use the twitter dataset with 11,000 documents or choose data from some source that allows you to download a large quantity of data. There should be thousands of documents and several property / value pairs for each document. Each document must contain at least one property having at least as much text as a standard tweet (140 chars).

If you use a different dataset, it should be easy for the instructor to import it into MongoDB using a file you supply. Do not require the instructor to access a remote copy of data by providing a URL.

Frontend Application

The user-facing portion of the application should have a web interface, e.g., an html document. The application should be developed in JavaScript or Python. The instructor should be able to run the application on the Ubuntu VM available on Kelvin. If extra packages need to be added to the VM, those should be specified in an accompanying `readme.md` file, written in markdown format. All submissions should include a `readme.md` in the style of github's `readme.md` files. The `readme.md` file should introduce the application to a complete stranger, as well as documenting any installation or running requirements. The instructor should not have to rely on any verbally communicated information to test the application. If the application can not run on the Ubuntu VM available on Kelvin, you will receive a zero for the entire project. If it does not run because you were not able to test on the VM because you somehow could not get the VM to work, you will receive a zero for the entire project. If you are still struggling to install the VM on the day of the demo, you should expect a zero for the entire project.

Functionality

For this application you should perform the following functions.

1. Connect to the database and show the connection process at startup. Each step in the startup process should have a message sent to the console or GUI. This includes the connection to the server, database and collection.
2. Present the user with web-based display that asks the user for a query string. It must support partial words. Examples

include but are not limited to “Foo” should find “Food” or “work” should find “piecework”.

3. The response to the user’s request should present the user with a selection list of documents answering the request, but not the content of the document.
4. Once presented with the selection list, by title, by date, or other identifying information, the user should be able to select the desired document to view the content.
5. The user is presented with the option to search again and again until they exit the page.
6. The user has the ability to add a separate annotation (e.g., a comment) to the selected document. Annotations are stored with the document and will show in subsequent retrievals of the document.

See the grading rubric for additional requirements.

Submission

Your submission to the MyCourses dropbox should consist of any program files, along with with any html files, css files, and a `readme.md` file. If you choose a dataset other than tweets, you must supply a file suitable for import into MongoDB.

Peer evaluations should be submitted to the “peer” dropbox in MyCourses.

Demo

You will demo your application to the instructor and, time permitting, to the entire class. The demo will be very simple.

1. Show the startup and attendant messages.
2. Show a query, result list, and viewing of one document.
3. Repeat but with a query chosen by the instructor or a student outside your group.

tweets dataset (20 points).

- Database on Ubuntu VM is sufficient as is for demo
- No credit if the database does not work based on the written instructions

Alternative dataset (20 points plus bonus points).

- Database is built and performs acceptably
- Database has sufficient number of documents
- Database has sufficient number of property value pairs and sufficient text
- No credit if the database does not work based on the written instructions

Program Functionality (60 points total 5 each).

- GUI solicits required information from the user
- Literal word search works acceptably
- Partial word search works acceptably
- Document list is returned to the user for selection
- Document list includes date and author
- User can select the document to view and entire selected document is displayed
- User can exit the view to return to the list
- User can exit the list to perform a new search
- User comments can be added to the documents
- User comments are retrieved with the documents
- Source code style is readable
- GUI appearance is acceptable

Demo (20 points total 5 each).

- Demonstrators can answer questions
- Demonstration shows startup

- Demonstration shows team-selected query
- Demonstration shows instructor-selected query

Total Points. 100 plus any bonus for alternative dataset

Note: If your program fails to run at all, you will receive a zero for the entire project. If the written documentation is insufficient to assist the instructor to install and run the application, you will receive a zero for the entire project. If the instructor must rely on any verbal instructions or information, you will receive a zero for the entire project.