**COMP 4620 Web-Based Information Systems**

**Seminar/Lab 7**

**Access to MongoDB for User Management**

1. **Objectives**

* Understand how modules work
* How to access MongoDB
* How to use Node WebServer to run server-side JS programs

1. **User management for ChatApp**
   1. chat\_view.html

* You can download the code from the Moodle and use it. You may need to change the variable, url, for the server-side program. Nothing else needs to be changed.
* There are three menu items for Join (SignUp), SignIn (Login), Unsubscribe

Graphical user interface, text, application, email

Description automatically generated Graphical user interface

Description automatically generated

* Three different POST queries will be submitted to chat\_controller.sjs.

Join: {command:"join", username:..., password:..., email:...}

SignIn: {command:"signin", username:..., password:...}

Unsubscribe: {command:"unsubscribe", username:..., password:...}

* 1. chat\_server.sjs
* This is the main server-side program.
* Three different POST queries will be passed to proceed() in the program.

Join: It should be completed only when the ~~username does not~~ **~~exist~~**~~.~~

SignIn: It should be completed only ~~when the document of the given username and password~~ **~~exists~~**~~.~~

Unsubscribe: The document having the given ~~username should be~~ **~~removed~~**~~.~~

* The return values through callback function are “Success”, “Fail”, or “Error”. Or similar messages.
* The query operations should be implemented with the functions in chat\_model.sjs.
  1. chat\_model.sjs
* This file is the one in “How to solve an example problem: ChatApp” in 4.3.10
  1. Note that chat\_view.html and two skeleton files of chat\_server.sjs and chat\_model.sjs can be downloaded from the Moodle. You can start with them.

1. **How to test?**
   1. Run **your Node WebServer**. (You may download the solution of Node WebServer from the Moodle into a different directory and run it with your port number.)
   2. Open chat\_view.html from a web browser, and test the three UserManagement queries.
   3. (You may try <http://cs.tru.ca:8080/~mlee/comp4620/Winter2022/9.%20Seminars/w7_chat_view.html> to see how chat\_view.html works.)
2. **How to keep running ‘node’ when you logout?**
   1. You need to run your Node web server using the ‘nohup’ command so that your web server can be tested later. For example,

$ nohup node index.js &

* 1. How to kill the above node process later?
* Find your process. E.g., $ ps -aef | grep node
* $ kill -p *process\_number*

1. **Assignment**
   1. Submission

* The URL of an HTML file that uses chat\_view.html in 2~3.
* Total marks: 10
* Due:
  + 11:59 PM, March 10, 2022 – 10% bonus
  + 11:59 PM , March 14, 2022 – full marks
  + 11:59 PM, March 15, 2022 – 10% penalty
* Any late submission will not be accepted.
* You should not update the application after you submitted. The information of any update of a file/directory will be left on CS, and I can check the last update date and time.
* The evaluation will be usually one of pass, half-pass, and fail. Most incomplete applications will get fail.