

Assignment 2

Chat Room Requirements

COMP4021
Internet Computing

Course Learning Outcomes

By the end of this course, you will have the following skill set.

1. General Appreciation

- 1.1. Have a general appreciation of the use of the Internet in society
-

3. Server Based Skills

- 3.1. Be able to install and understand the operation of a server such as Apache
- 3.2. Develop server side code in an appropriate language such as PHP

4. Skills Related to Both

- 4.1. Have a working knowledge of the most common HTTP instructions and their methods of client-server interaction, including cookies
- 4.2. Understand XML and related technologies including DOM handling
- 4.3. Develop complex programs for browser-server communications, including use of Ajax

Course Outcomes (Reminder)

2. Browser Based Skills

- 2.1. Be knowledgeable about HTML and related display techniques including CSS
- 2.2. Understand how to build browser based programs using the JavaScript language, including DHTML and event handling
- 2.3. Be able to program advanced browser display technologies including Flash and SVG, and to appreciate the differences
- 2.4. Develop code for handling communication between web page components such as JavaScript.

Overview

- 16 marks – Chat Engine
- 16 marks – Logging In
- 18 marks – User Name Checking
- 16 marks – Logging Out
- 16 marks – Chat Message Input
- 18 marks – Automatic Hyperlink

Max Mark = 100

Bonus

- Use `getUserMedia()` to record audio in the client side and upload as chunks to the server
- 30 marks
- https://github.com/imalhasaranga/Html5_Video_Audio_Recorder

16 Marks – Chat Engine

○ Basic chat system issues

- Correct system procedure
- Appropriate transmissions from client to server
- Appropriate transmissions from server to client
- I.e. Automatic update of the chat room when someone types a message
- And so on

Technologies Used

- ◉ Client side skills: JavaScript, Applet, AJAX, HTML forms, event handling
- ◉ Server side skills: PHP
- ◉ Both: Cookies/sessions, XML, DOM handling, HTTP

16 Marks – Logging In



Please enter your user name:

- A user must enter name to enter the chat room
- An error will be shown if any of the information is not correctly entered or selected

18 Marks – Username Checking



- A username can only contain letters and digits
- An error will be shown if any of the information is not correctly entered or selected

16 Marks – Logging Out

- There should be a 'log out' button somewhere
- The 'button' could be anything you want – SVG/HTML, etc
- All the chat messages on the screen and xml file are cleared, and the browser returns to the log in page

16 Marks – Chat Message Input

- User can enter his/her chat message in a text field
- Once the text message is sent it is shown as the latest message on the chat room

18 marks – Automatic Hyperlink

- If a user enters a URL within the text of the chat message the URL is converted to a hyperlink automatically
- We assume that a URL is a substring in the text which starts with 'http://' and does not contain any space
- For example,
 - "Please look at http://google.com now!"



This is a URL



Bill: Please look at `http://google.com` now!

Your code converts ‘`http://google.com`’
to a hyperlink automatically



Bill: Please look at `http://google.com` now!



A clickable link

- If a substring is a URL it is converted to a hyperlink so that:
 - The hyperlink is displayed as an underlined text
 - The user can click on the hyperlink to display it in a new browser window (instead of replacing the current page, i.e. the chatroom)

Make Lots of Backups

- Keep backups of your work
- Some possibilities:
 - Ftp a copy of your files to *ihome.ust.hk* and set permissions so only you can read them
 - Email yourself the latest file(s)

Single Person Work

- This is a single person work
- We'll check your work using the configurations given in the lab pages(WAMP server)

Work in *www*

- All your work must be developed in the *www* subdirectory inside the WAMP directory (or *htdocs* subdirectory inside the Apache directory), i.e. you can use *<http://localhost/chatroom.html>* to access the chat system on the server
- All project files in the directory must be submitted
- If you use sub-directories that's OK, as long as they are sensibly used

Dump Unnecessary Files

- Before you zip and submit your work, remove all irrelevant files in *www* (or *htdocs*)
- For example, dump the example *index.html* files, etc

Submission and Deadline

- Put all the *www* (or *htdocs*) files into a single zip file called
Name_StudentID_comp4021_a2.zip
- Make sure you include any sub-directories
- Upload the zip file using the CASS system before the deadline
- The deadline is **Monday, 4 May 2013, 23:59**