IN712 Web 3 Semester 1

# Assignment 2 – node.js

# Group Size: 1

# Value: 25% of course mark

# Due Date: Friday, April 14th, 5.00 pm

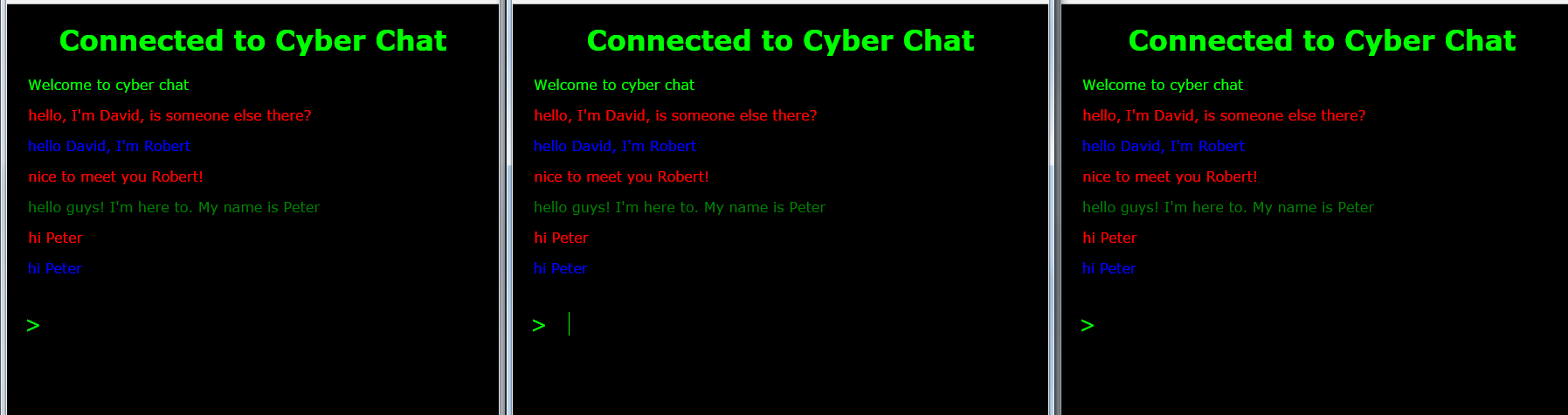
## Requirements (each of the requirements (bullet points) is worth 1 mark, partial marks will be given for partially completed tasks):

* + Using node.js, create a Web server that is able to serve the html file from folder public (i.e. Do not use any frameworks!!!) as provided in the assignment resources located in the I drive/GitHub repository.
  + Implement the functionality that allows your web server to be able to properly serve .css and .jpg files (also in folder public). Do not use any external libraries, do it in raw node.js (i.e. Do not use any frameworks!!!) . Use regular expressions to detect file extensions such as .css or .jpg in the request object.
  + Once you have detected a request for a given resource (.css, .jpg), write the appropriate Content-Type header on the response object
  + Use streams to serve the .css and.jpg files
  + Make your Web server only respond to requests on port 3000.

The final result when you hit <http://localhost:3000/> should look like the following:



* + Provide me with the online chat application you created in the web socket practical
  + Extend the functionality of said online chat application so that messages from different clients are color-coded in the chat for easy identification of who is saying what. An illustration of the functionality you need to implement follows, displaying how the application should look on 3 different client computers (basically, it should look the same in all of them):



* + Create a web application using the Express framework for node.js that displays information about famous scientists using the URL scheme showed in the video demo.
  + Make sure your application contains a package.json file to handle dependencies
  + Your application should store on the server the data about the scientists using the JSON format
  + Your application should allow users to submit suggestions for inclusion in your scientist Hall of Fame and that information should update the JSON file on the server containing the scientist information.

Try to mimic as closely as possible the behavior of the application I have created and that I demo in the provided video e3Demo.avi in the I drive/GitHub repository. Pay particular attention to using the correct URLs routes as shown in the video demo. When testing the uploading of information to your web application, you can use the following web resource to get information about famous scientists <http://listverse.com/2009/02/24/top-10-most-influential-scientists/>

* + Deploy your best scientists application to Heroku, and in your assignment folder provide me with a readme file containing the live URL where your application lives so I can check it out.
  + You want to create a website (imaginarily located at <http://thebestscientisteveeeeeer.com>) that ranks famous scientists according to the number of citations that their published works have received over time. You decide that you would like to use the resource “Google Scholar” (see <https://scholar.google.com/citations?user=SfDzdgEAAAAJ> for an example) to fetch dynamically the number of citations of each particular scientist using the XMLHttpRequest object from your JavaScript client code sitting in the domain <http://thebestscientisteveeeeeer.com>. Would you be able to do this?

Note that this task is theoretical in nature. You just need to write a comprehensive answer containing the (HTTP based) justification for what you think it would happen if you attempt such a trick. You don’t strictly need to implement any code. Although you could in order to get hints about whether you can do it or not. Create a txt readme file in your folder for exercise 4 and write on it your answer.

* Create your own private repository for you to work on this assignment by using the following link: <https://classroom.github.com/assignment-invitations/2b6dfc619be3db092abb1dafc14b77ff>
* Your code must be modular, readable, scalable and robust.
* Your code should be properly documented
* All code files and resources in a legible file structure should be available in your own private GitHub repository “https://github.com/OPClasses/assignment2-YourGithubUserName”.

## Learning outcomes:

1. Understand the structure of a compiled web application or webservice and roles of elements in that structure
2. Understand web application, webservice, web pages and web control lifecycles and design items accordingly.
3. Understand and appropriately use caching, master pages, Asynchronous JavaScript and XML (AJAX), software service providers, datastore interaction, Cascading Style Sheets (CSS).
4. Understand and appreciate the impact of Web 2.0 and the semantic web.
5. Implement web software designs using an object-oriented language in a contemporary integrated development environment