Assignment 3 (Due: March 23 midnight)

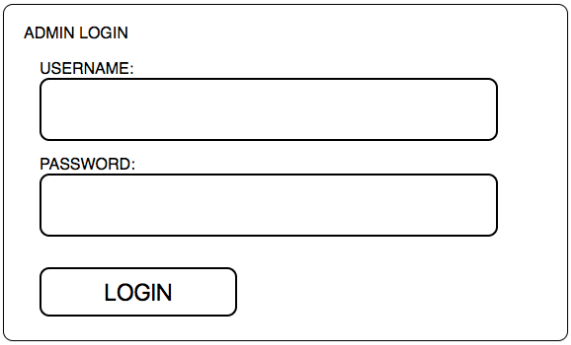
In this assignment, we would like to implement a simple online check-in application. We will be using the npm packages Express and Mongodb.

DESCRIPTION:

The check-in application will behave as follows:

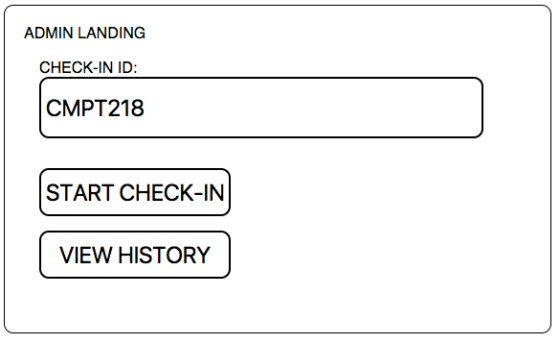
1. When a client sends a GET request for the "/" folder (the root of the application), they will be served a form called “login.html”. The page will contain two sections:
   a. A form containing fields for the admin user’s login. This will include the username and PIN number. The admin user has the credentials:
      Username: admin
      PIN: 1234
      For this assignment you may hardcode this into your application. For obvious reasons, this will not be a good solution going forward.
   b. A link that says “check-in now” that will redirect the user to a “check-in” page (see point number 3 below).

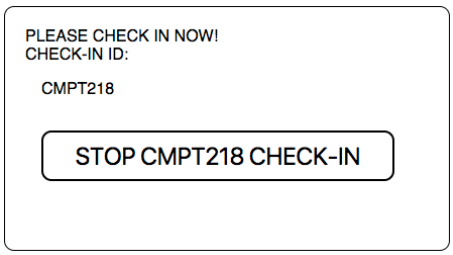
   One possible layout for the form is on the right.



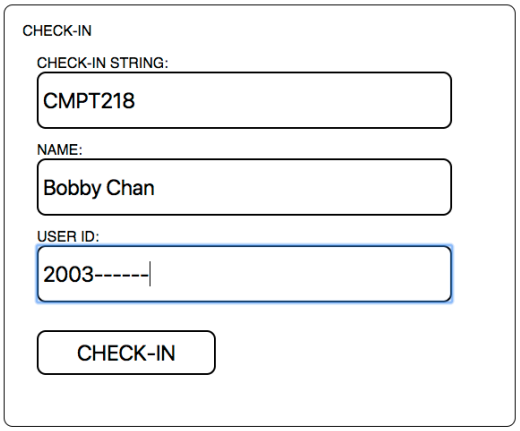
CHECK-IN NOW

2. When the admin login form is submitted with the correct credentials, the user should be redirected to an “admin landing” page (although we’re dealing with a very limited amount of security here, you may want to generate this page dynamically just so that users cannot just get to this page by requesting a GET on the page). On this page, the admin user will be able to have the ability to enter a string (for our purposes, let’s assume it is a course id) and “start a new check-in” or to “view past check-ins” corresponding to that string. For example, in the example shown below:
   - If the admin user clicks on “start a new check-in”, a new check-in object with check-in id “CMPT218” should be initiated on the server and the user will be redirected to a page with a button that says “stop CMPT218 check-in”. This is when attendees can “check-in”. After clicking “stop cmpt218 check-in” the user will then be redirected to a page showing the total number of attendees and all attendees who have checked-in. This information will also be written into the database as a document.
   - If the admin user clicks on “view history”, they will be redirected to a page showing all CMPT218 checks recorded in the database, including all attendees in each check-in.





3. From the original “login.html” page, if a user clicks on the “check-in now” link, they will be directed to a “check-in” page where they can enter the check-in string, their name, and user ID number (for our purposes, let’s assume this is the student number).



When the user clicks “check-in” at this point, this attendee will be added to the server object with check-in ID “CMPT218” and redirected to a page that says “Thank you for checking-in!”.

ADDITIONAL NOTES:

You are allowed to add any other files you’d like (i.e. a css file or a js file). Once again, it is totally up to you how you would like to organize your code, but here are several requirements:

- Implicitly stated in the description, you must use a mongoDB database.
- This description is the minimal requirements. You can assume that only one check-in event is happening at one time (for example, there will not be a “CMPT218” and a “CMPT222” check-in happening at the same time).
- It is up to you what information you would like to keep as a database schema. I would recommend that you keep a date attribute for each of the check-ins to differentiate between (for example) CMPT218 on Feb 25, 1:00pm and CMPT218 on Feb27, 2:00pm.
- You may use the NPM packages Express and Mongodb or Mongoose. You should not need anything more (especially the packages that already implement check-ins :P).
- Make your code as robust as possible, meaning that you can move the whole application somewhere else and it will still work :)
- Usability: it is up to you how you would like to make your app more usable. Some suggestions are to add a delete check-in functionality and allow multiple check-ins to occur at the same time. Remember the purpose of the application to take attendance at an event :)
- For this assignment, you may use client-side frameworks such as jQuery, Ember, Angular, bootstrap, etc.

As always, if you are unsure if something is allowed, please ask either Bobby or one of the TAs.

Assignment 3 (Due: March 23 midnight)

Marking Scheme:

coming soon …

Submission:

You code and app URL should be uploaded to coursys and a running application should be present on cmpt218.csil.sfu.ca:<yourPort>.

Please note

1. it is very important that you only use the port that is assigned to you. Please double check this before you submit your assignment otherwise, your grade will be 0.
2. do not make any changes to your app after the due date, we will check your app's last modified date before marking