CSCI2720 Group Project Specification (2-3 students per group) **Due date: 15 Dec 2017**

Objective

Implement a *token redemption* web application to allow

- users to redeem tokens for some intangible items
- the web application administrator to manage listed items

Characteristics of the Data

A) Item

An *item* has the following attributes

1. **ID:** A unique value that identifies the item.
2. **Title**: Plain text
3. **Description**: Formatted text
4. **Image**: At most one image
5. **Token value** : Integer (# of tokens needed to redeem the item)
6. **Available Quantity**: Integer
7. **Tags**: Zero or more strings (plain texts)
8. **Creation timestamp**: A value to indicate when the item was added to the system

“Formatted text” means a subset of HTML that includes only elements “b”, “i”, “u”, “pre”, “p”, and “br”.

B) User

A user must have at least the following attributes:

1. **Username**: A unique string that contains 8-12 alphanumeric characters.
2. **Password**: 1-way hashed value of the user’s actual password
3. **balance**: An integer (# of token left in the user’s account)

Required Features

1. **An interface to present the detailed view of an item**
   - All the attributes of the item should be shown on this interface.
   - On this interface, an authenticated user should be able to redeem the item if the user has enough tokens and the item is still available.

2. **An interface to list items**
   - The interface should implement “pagination” and list at most 10 items per “page”.
   - Through the interface, a user should be able to change the order in which the items are listed. In particular, the user should be able to list the items in non-decreasing/non-increasing order based on the item’s
     a. Creation timestamp
     b. Redeem cost
   - For each listed item, only the item’s title, token value, available quantity, and creation time should be shown.

- When the item's title is clicked, the detailed view of the item should be presented to the user.

3. **An interface for an authenticated user to list all the items he/she had redeemed.**

- For each listed item, you only need to show the item's ID, title, token value, and the timestamp at which the item was redeemed.

4. **An interface to allow the administrator to perform CRUD (Create, Retrieve, Update, Delete) operations on the items.**

5. **A way to allow the administrator to export a list of redeemed items as CSV file.**

- For each item, you only need to show the item's ID, title, token value, the timestamp at which the item was redeemed, and the username of the user who redeemed the item.

- You should use a proven 3rd-party module to produce the CSV (Comma-Separated-Value) file to ensure it is encoded properly.

- Your application should prompt the user to save the exported file instead of showing the file content in the browser.

6. **User authentication (Log in)**

   The application should provide a way for user to authenticate his/her identity. That is, to login via username and password.

7. **For authenticated user, the application should always show the user's username and his/her token balance at the top of the interface.**

Other Requirements/Assumption

- You web application should run on Node.js.
- Your application DOES NOT need to support "user registration". You can hardcode the user data (including the token he/she has) in the database.
- Feel free to populate the database before demonstrating the application to the teacher.
- You may use MySql or MongoDB as your database.

Submission and Project Demonstration

- Please prepare a ZIP or ARJ file containing the following files and submit the ZIP/ARJ file to eLearn:
  - A folder containing all the files that make up your application.
  - A README document with the following information:
    - Assumption that you have made
    - Instructions to setup the database so that your app would work properly if the teacher run "node ." in the root folder containing the app
    - List of features that are not implemented in the app
    - Known bugs
  - Other files (e.g., scripts) that you use to populate the database or test your app, if any.

- Please arrange with the teacher a time on or before the due date to demonstrate your application.

Assessment

1) **Required features (60%)**
    - Do all the required features implemented and function correctly?

2) **Is the application fully tested? (20%)**
    - Had the application been thoroughly tested (for all possible input)?

3) **Design (20%)**
    - Is the user interface intuitive?
    - Is the application easy to use?
    - Does it look good? (You are encouraged to use 3rd party CSS such as Bootstrap, Materialize to style your application)