Assignment 1 (Due: April 17 noon – **please note HARD DEADLINE**)

In this project, we would like to implement an application to allow the user to upload a file of final percentages for a course. The application would then allow the user to assign grades while displaying a histogram of the results.

Requirements and Workflow

Login:

The root of your application should present a login form to the user. Login validity should be checked against a 'users' table in a database with the following credentials:

MySQL Database:

- Host: 35.227.146.173
- Database: cmpt470
- User: readonlyuser
- Password: readonly

The database contains a table with two columns for a list of users along with their md5 hashed passwords (with the attributes username and password). As an example. If the valid users are

username	password
bobby	1234
steve	hello

Using the md5 (unsalted) hashing algorithm (https://en.wikipedia.org/wiki/MD5), the table should be stored as

username	password
bobby	81dc9bdb52d04dc20036dbd8313ed055
steve	5d41402abc4b2a76b9719d911017c592

Currently, these two users have been added to the users table. When we test the application, there will be other names, so please make sure you do not just hard code these values.

Invalid logins will redirect you back to the login page. Each subsequent page will contain a logout button for users to erase their current session.

Landing page:

Once user has logged in, they are presented with a page to upload a csv file. This file corresponds to grades in a particular course. It has the headings **studentID** and **all grade items in a semester**. In particular, there is one row whose student number is called 'total' which indicates the percentage weight of the grade item for the semester; the total row could be anywhere in the CSV file. Your application should read in the csv and translate it into programmable objects. A sample CSV file might contain the following info:

studentID	quizzes	midterm	final
asm23	83.45	56.7	77.4

Assignment 1 (Due: April 17 noon – **please note HARD DEADLINE**)

bobbyc	66.7	78.22	34
hermr	93.5	86.5	89.02
dude32	23.4	12.33	34.87
total	20	35	45

The total of the semester should sum to 100. Otherwise, the application should prompt the user to upload another file.

Histogram generator:

The user will then be directed to a histogram generator that will allow the user to choose cutoffs for letter-grades corresponding to the student's final percentage (this must be a different file/route). As the user selects the cutoffs, a live histogram should be generated to show the number of students in the range of each letter-grade. A cutoff is the percentage between letter-grades. I would recommend sliders with error checking. An example of a histogram:

```
                         O
     O    O              O                   O
     O    O    O         O    O         O    O    O
O    O    O    O    O    O    O    O    O    O    O
A+   A    A-   B+   B    B-   C+   C    C-   D    F
```

Please note this is just a suggestion, you can be more creative here. After the user sets appropriate cutoffs for the letter-grades, he/she is directed to a printout of the final results for each student including studentID, grade items, final percentage, and a letter-grade.

Assignment 1 (Due: April 17 noon – **please note HARD DEADLINE**)

Architecture and Technical Requirements:

Docker

Your whole application must reside on one docker image that you must push to a docker repo. The docker repo can be public or private (adding the TA and I as a collaborator). If you are making it public, I highly recommend that you choose a name and tag that is not easily guessed. We will be running the command

```
  docker run -p 12345:8080 -d <your ID>/<repo>:<tag>
```

and testing your application on `localhost:12345`. It is up to you to make sure that this works.

Application Code

Your application code should be in a folder called “/app”. You do not need to follow any specific application architecture as long as the code works. Please note that I will be checking this folder against your submission (and also against other students’ submissions :P).

Other technical Requirements

You may use NodeJS packages or PHP libraries with these exceptions:

1. You must implement the histogram yourself without the help of third-party packages or libraries.
2. You are to implement a login by implementing your own sessions (login from scratch). You may not, for example, use packages/libraries in Laravel (PHP) or Passport (NodeJS).
3. You may not use any front-end JavaScript frameworks (such as React, Angular, etc) nor Bootstrap. JQuery is ok since there may be operations that can only with jQuery ☺.

UI

It is totally up to you how you’d like to design the UI but please keep in mind that usability and creativity will be a small portion of your mark.

Assignment 1 (Due: April 17 noon – **please note HARD DEADLINE**)

Marking Scheme:

5 marks: Login using own sessions
5 marks: Reading CSV files and turning it into objects
8 marks: The correct histogram should be calculated dynamically given user inputs.
2 marks: results displayed to user
3 marks: Creativity and Usability. Be sure to check for error cases and exceptions.
3 marks: Proper submission - Dockerizing the container properly. See submission below.

26 marks: **TOTAL**

Submission

You should submit the following onto coursys.

- **A readme file** including your docker repo identifier <your ID>/<repo>:<tag>. If you were not able to complete any of the parts, please indicate what you were able to complete.
- **Your code**. This should also match the code stored in the /app folder of your docker image
- As mentioned above, We will be running the command

  ```
  docker run -p 12345:8080 -d <your ID>/<repo>:<tag>
  ```

  and testing your application on `localhost:12345`. If you've created a private repo on docker, please add the following docker IDs as collaborators:
  - bobbyctchan
  - sumukhab