# CS260 - Perl
# Final Project – Fall 2016

**Objectives**
Create a script-based website
Work with CGI scripts

The final project will implement a system that stores the class you have taken at UND. The basic functionality of the web-based scripts will be to list all the classes you have taken, search your classes by name or department, or add or edit a class. All search functions should be available to all users but a user must login before being able to add or edit a class.

The info you need to access the database

**Database name:** f16final
**Login:** root
**Password:** password

Here is what is in the f16final database. The database is set up as follows:

**Table:** `tblclasses`
**Fields (and in order but that shouldn't matter):**
```
classid – autoincrement integer
classname – text – maximum 100 characters (Computer Science I, Computer Science, II,)
department - text - maximum 5 chararacters (CSCI, MATH, PHYS, EE, ...)
classnum – text – maximum 5 characters  (160, 161, 260, ...)
grade – text - 1 character (only options are A, B, C, D, F, S, U)
credits – integer
```

**Table:** `tblusers`
**Fields (and in order but that shouldn't matter):**
```
userID – autoincrement integer
login – text – maximum 50 characters
password – text – maximum 20 characters
name – text – maximum 50 characters – this will be the users "real name"
```

**Specifics**
**Main page** - There must be a main page should allow the user to login, display all classes taken, or search for a specific class(es). If logged in, the user should also be able to add a class or edit an existing class. This page must be named index.pl or index.html.

**Login page** - The login page will ask for a login and password. Check that information against the `tblusers` table. If the login and password are valid, set a cookie indicating that the current user is logged in (and therefore legal to add/update classes).

**Transcript page** - This page will list classes taken, either displaying all of the classes or after a search. After the classes are listed display the total number of passed credits, attempted credits, total number of honor points and the grade point average with exactly 3 decimal places FOR the classes displayed. Classes with a grade of S or U will not count towards the gpa but towards the number of attempted credits. S grades will also count toward the number of passes credits. If a search is to be used the user must be able to search by either the class name or the department. Matches do not have to be exact. The search can be implemented in a separate match or on the display page. If the user is logged in include options for adding or editing a class.

**Add a class page** - This page will accept all the information for a class (other than the class id) and add it to the database. There is minimal error checking of the information that will be required. All fields *must* have a value (meaning I won't try to put string data into an integer field, but I will check to make sure all fields to have a value) and credits must be between 1 and 12 (yes, I believe you can take a 12 credit class). An option to leave the add form without actually updating the database is required.

**Edit a class page** - This page will populate the fields with the current info (other than the class id) and then update the existing record in the database. This page has the same error checking as in the add scripts, there is no specific error checking required other than to make sure all the required fields have values. An option to leave the update form without actually updating the database is required. It's likely this page and the add book page can be accomplished with the same script.

**Notes:**
ALL scripts that allow for changes to the database will need to check and see if a valid user if logged in to prevent an unauthorized user from jumping directly to a script in which they do not have permission. In that case, display a simple message indicating the user does not have permission to run that script. And a link to the main page (or login page) would be nice.

Any issues regarding plagiarism will result in an immediate incomplete grade for **all parties** involved until the matter is resolved (which is likely to be at the start of the spring semester). Once the issue has been handled, if plagiarism is found to have occurred **all parties** found to be involved will receive a grade of 'F' for the course and may be reported to the Dean of Students for academic dishonestly.

While I welcome the use of HTML tags, color, and such, it is not a substitute for the requirements of the project. I do expect the information presented will be easy to read, I do not expect a professional layout. Make use of tables and other simple formatting such as bold/italics, colors, centering and fonts.

Keep the time frame for logged in users to something short, say 1 minutes or even 30 seconds. Users should stay logged in as long as they are active in the website.

**Show and tell - this may be altered into simply turning in the project, I'll let you know in class**
All students will be required to demonstrate their project. Each student will be allocated 30 minutes. You will be asked to demonstrate the basic functionality of your project. You can bring a laptop or a flash drive with the files. A sign-up sheet will be placed on my door. Fill your name in any open time. **I** am the only one that can remove a time, including a time you have entered yourself. Once you enter a time into a slot it's yours until **I** remove it - this makes it easier to ensure no one can hijack someone else's time slot.

**Turning in the project**
A single zip/tar file is required. The file name MUST be your first initial followed by your last name and the phrase '260final'. My file would be tstokke260final.zip. In the compressed file there should be an HTML folder (if necessary) and a PERL folder. Place your HTML files and PERL files in the appropriate folder. If you use subfolders, please include those folders as well in the compressed file. When testing I want to copy files directly from your HTML/PERL folders to my HTML/PERL folders on my test machine. The "starting" webpage MUST be named index.html or index.pl; all web pages must be accessible, directly or indirectly, from index.html or index.pl.

**Questions?**
I've outlined the basics. If you have questions ASK. I've kept this open enough to allow for flexibility when designing the web pages.