CCCS 310
Course Project
Due: Last day of class

Step 1: Get your team proposal accepted by the professor (No later than March 1st)
- Team of three (ideal)
- Select a team leader to communicate with the professor
- Add your team to the Google doc

Step 2: Make an appointment for your presentation (No later than March 14th – email me)
- Appointments will be scheduled backwards. I'll start scheduling from the last class unless you specifically ask for a date.
- Each presentation must be 15 minutes long. This means a maximum of 4 presentations per class. All your team members must speak. I will stop you at the 13th minute, finished or not. You may be tempted to speak quickly – don't. Better fewer slides and slower talk. Your presentation is 12 minutes long including a demo, plus 2 minutes for questions, and one minute for setup.
- You are expected to (1) Have a title slide with your project name and your team member's names, (2) describe what your application is about, use 1 slide. (3) describe the architecture you used: one slide for front end, one slide for back end, one slide for special technology or techniques, one slide on what you learned positive, one slide on what you learned negative. (4) You must demo your application. (5) Questions.
- Diagrams are often better than text in this type of presentation.
- You must email me your presentation. You will be using my laptop for your presentation.

Step 3: Start programming
- The minimum requirements for the project are:
  - a <u>modern</u> website
  - uses a <u>minimum</u> of 3 Internet technologies per teammate.
    - Technologies are evaluated through a point system:
      - HTML5/CSS/CGI .. .. 2 points
      - JavaScript/DOM .. .. 2 points
      - JavaApplets 1point
      - XAMPP .. .. 2 point (using the servers from COMP 206 do not count for points),
      - PHP .. .. 1 point
      - MySQL .. .. 2 point
      - JSON/XML/text .. .. 1 point
      - React .. .. 1 point
      - Java Servlets .. .. 1 point
      - Python/Perl/C/Bash .. .. 1 point
      - socket programming .. .. 2 point
      - Security .. .. 1 or 2 points
      - You may suggest others .. .. ?? points
  - If your application stack is all one language, like JavaScript, then I need to look into this carefully. Mix something into it, like SQL and HTML5.
  - HTML5 means using the version 5 features, or it does not count.

Step 4: Submission

- A readme.txt file with your team member names
- An HTML file that links directly to your website or an easy download of your application
- A ZIP of the back end source and databases/files
- A ZIP of the front end source and databases/files
- Instructions on how to run your website
- All team members submit the entire project (so that the grade box is open)

Step 5: Demo during your presentation

Example good project structures:

Standard website (best learning outcome)

FRONT END	BACK END
HTML5+CSS+CGI	mySQL
JavaScript+DOM	Security
JSON	PHP
	Apache

Tool driven website

FRONT END	BACK END
JavaScript+DOM	NodeJS
HTML5+CSS	SQL
AJAX	Security
JQuery	

Stand-alone Social Application (no website for the user to access directly)

FRONT END	BACK END
C	SQL
GUI	Python
Sockets	Security
JSON	XAMPP
Encryption	

HOW IT WILL BE GRADED

- 100 points (assumes 3 members in a team)
- +10 points for a fully running website
- +10 points per technology
    - Full points if the technology was used correctly and not in a trivial way.
- -10 points for not following instructions
- Points are awarded proportionally
- Teams with less or more members will be scaled accordingly