Website and Web Application CSc 473

Syllabus, Spring 2015

Website and Web Application CSc 473, Syllabus, Spring 2015

General Information

Instructor:	Michael D. Grossberg
Email:	grossberg@cs.engr.ccny.cuny.edu
Office:	NA 7/311
Office Phone:	212-650-6166
Office Hours:	Tues 3:30pm-4:30pm, Thurs 9:30 am-10:30pm
Website:	Course web page hosted on cuny blackboard https://bbhosted.cuny.edu

Course Meetings

Location:	NA-6/268 moves to NA-7106.
Time:	T,TH 11:00PM - 12:15PM

Course Description

From Bulletin:

> The design and implementation of web sites from a Human-Computer Interaction viewpoint, with emphasis on user testing. Navigation design. Accessibility by persons with limitations in vision or motor ability is stressed and must be addressed in the final project.

More accurate:

> The design and implementation of web sites and web applications. Current web technologies will be reviewed as well as principles of user experience design. Students will learn to write a web application in a web framework. There will be an emphasis on testing, working in a small team and software engineering best practices.

Note

This course will assume some programming ability. If you have never programmed before, you will need to allocate time for self-study. This is a CS course, not a graphic design course so there will be a significant component of server side programming. Some people assume that web programming is "easier" than application programming. It is **harder**. You are expected to use many technologies such as HTML, CSS, Javascript, Python, XML, YAML JQuery, and many APIs. **Not** covered but also used technologies include Java, PHP, Ruby on Rails, Flex/Flash, Silverlight/.Net, and SQL. Lectures can only provide superficial overviews of the some of the technologies you will need. You will be expected to do significant independent study. While challenging, comfort with these skills are in great demand in the workplace.

Collaboration and Academic Integrity

It is acceptable, even encouraged to form study groups and collaborate in understanding homework problems, and preparing for exams. However, **all** the work on homework and exams should be your own work exclusively. It is very easy to test for duplicate code. Some students do not realize that professors can use Google too! Checking that two files have similarities is easily done with Unix tools like grep. Cheating will not be tolerated. The work of a group project must be done exclusively by members of that group.

Cuny Policy on Academic Integrity

Grades

The components that will determine the grade are

Weight	Component
20%	Homeworks
15%	Midterm
40%	Project
25%	Final

Homework

Homework will be submitted online. You will put your homework in a bitbucket repository and share it with me. **Do not** send by email; it may be lost.

Midterm/Final

Will consist of some questions that test the material by direct questions on languages or technology, and questions which ask you to write code or web pages (at a computer).

Project

You will participate in a group project to build a functioning live web site. Your **individual** participation in this project will be measured by the number and quality of the code/documentation you write as measured by the version control system. If it is not clear from the checkins that you contributed it will be assumed your team-mates did all the work.

!DANGER!

The *ONLY* work that is counted toward your project grade are explicit working lines of code committed to the **master** version control system, issues and comment issues in the master issue tracker, meaningful edits in the project wiki, and quality of the final demo and presentation. Students often wonder why team members get higher grades for the same project when they "did more work." If the work isn't tracked within the project tracking system, it doesn't count. You are expected to commit small amounts of working code regularly.

If you commit large amounts of code rarely, you may lose points. Similarly if you try to make your commits look large by making many tiny inconsequncial edits, your commits will be scaled down. I will randomly sample your code and if I see such a problem I will scale accordingly.

Textbook

- No textbook.

The following are some of the references for the course:

> ***Note***
>
> Generally avoid W3 schools. At one time this was a very useful resource but not much of the material is out-dated, inaccurate and has seriously degenerated. Consider w3fools.com.

Version Control:

- Bitbucket
- Mercurial: The Definitive Guide
- hg tip Learn Mercurial one bite-sized tip at a time
- Mercurial for Beginners: The Definitive Practical Guide
- Tutorial on Using Mercurial
- Hg Init: a Mercurial tutorial

General Web References (HTML/CSS/JS)

- Learn Web Design
- Web Platform
- W3C Web Education Community Group Wiki
- Mozilla Developer Pages
- Opera Web Dev Pages
- Site Point

HTML/HTML5:

- Dive Into HTML5
- HTML5 Rocks
- Google HTML Video Tutorial

CSS/CSS3

- The 30 CSS Selectors you Must Memorize
- Mastering CSS, Part 1, Styling Design Elements
- Mastering CSS, Part 2, Advanced Techniques and Tools
- Google CSS Video Tutorial
- CSS-Tricks

Python

- The wisdom you are seeking is here, my child.

- Google's Python Class
- How to Think Like a Computer Scientist - 2nd edition
- Learn Python The Hard Way
- Dive into Python
- Python for Programmers

Google Technologies

- Google App Engine
- Python Runtime
- Video: Google Spreadsheet as DB
- Periodic Table of Google APIs

Other Technologies

- A ReStructuredText Primer
- Python Sphinx
- BeautifulSoup
- Python Bottle
- Django
- Pinax
- SASS
- Twitter Bootstrap
- AngularJS

Magazines and Articles

- Learn the Basics of Design This Weekend
- Smashing Network
- Six Revisions
- Reddit WebDesign
- Hacker News
- Web Designer Depot
- Web Design Ledger
- Design Float
- DZONE
- A List Apart
- Boxes and Arrows

Software Required

The official programming languages for the course are Python, and Javascript. You must uses these languages. You will need:

- A text editor Examples include Sublime Text, TextMate, TextWrangler for Mac OS; Kate or Gedit for Linux; jed, vim or emacs on all platforms
- A python 2.7 distribution (possible 3.4) you can download python from www.python.org, but *do not use IDLE*. For Mac OS there is Homebrew, and on linux the built in package manager will help
- Virtualenv (a python package) which comes with "pip"
- Mercurial, hg, get with "pip install hg"

Note

You are strongly encouraged to work in a Linux or Unix based OS like MacOS. If you are working on windows you can either dual-boot windows and Ubuntu or use virtual-box or VMWare. You exams will take place on Unix machines. Your web apps will be deployed on Unix machines. You MUST know your way around a Unix system. Working in windows may introduce trouble due to the difference in line termination. Just avoid the problem by working in Linux or MacOS.

Required Services

- Bitbucket Account You **MUST** get a bitbucket account. It is likely that you will need to upgrade to the $10/month account for the course. You may not share bitbucket accounts with a classmate.
- GoogleApp Engine You must get a google app engine account. You will probably not need to buy anything from google but you must set up the account.

Optional

- Heroku, Amazon EC2, Dreamhost, or place to host your project. Some will find the free GoogleApp Engine very difficult to work for Django and want a real database. I don't have a free provider for you but for a review of django oriented web hosts http://bit.ly/ULk7wa . It should be relatively inexpensive for a team given the time saved in frustration and the money saved with no expensive textbooks.

Topics Covered

Tentatively this is the plan of the lectures

- Version Control, Bitbucket
- Testing for the web in Python
- HTML5
- CSS
- WebFrameworks and URL Dispatch
- Google App Engine
- Web template engine (jinja2)
- Databases and Object Relational Mappers (ORM)
- Graphic Design Concepts, including Layout, Type, Color
- Requirements Process
- Usability
- Javascript, AJAX

- Web APIs